NetFusion-EXP Development-Kit

Reference Guide

Table of Contents

1 INTRODUCTION 8

2 ESTABLISHED DOCUMENTATION 9

2.1 BASIC NETFUSION CONCEPT, BENEFITS AND BRIEF 9
2.2 LIBERO SMARTFUSION2 FPGA FABRIC 9
2.3 HARDWARE PCB REFERENCE 9
2.4 FULL SOFTWARE & UCLINUX SUPPORT AND OPERATIONAL SUPPORT 9
2.5 NETWORK PERFORMANCE AND SPEED TESTING 9
2.6 RELEVANT REFERENCES 9

3 OUT OF THE BOX, DEVELOPMENT-KIT, POWERING UP & PROGRAMMING 11

3.1 KIT CONTENTS 11
3.2 APPLYING POWER 15
3.3 APPLYING POWER 15
3.3.1 Bench Supply 16
3.3.2 Screw Terminal Installation 17
3.3.3 Output 5V Auxiliary 17
3.3.4 JTAG Programming 18
3.3.5 USB UART Terminal 19
3.4 CONNECTING A CAT5 CABLE FOR TFTP UPLOAD 22
3.5 BUILDING THE NETFUSION CUSTOMIZED UCLINUX KERNEL 22
3.5.1 Building the SQLite, linuxptp, testapp & demo application incrementally 24
3.5.2 Building the main uCLinux with NetFusion Incrementally 24
3.5.3 Uploading the kernel to the NetFusion 25
3.5.1 Full uCLinux Documentation & Software Development 26

4 NETFUSION-EXP HARDWARE PCB 27

4.1.1 Net Fusion-EXP Baseboard Block Diagram 27
4.2 COMPONENT DESCRIPTION DETAILS 29
4.2.1 USB UART Interface 29
4.2.2 Ethernet Interface 30
4.1 USB OTG INTERFACE 31
4.2 FLASHPRO JTAG INTERFACE 31
4.3 ARM JTAG INTERFACE 32
4.4 BNC COAX SOCKETS 33
4.5 POWER SUPPLY 33
4.6 RESET BUTTON 34
4.7 GENERIC OPTION SWITCHES AND CONFIGURATION DIALS 34
4.8 USER BUTTONS & LCD 35
4.9 STEREO AUDIO PHONO-JACKS 35
4.10 TEMPERATURE SENSOR 36
4.11 VOLTAGE MONITOR 36
4.12 REAL-TIME CLOCK 37
4.13 RS232 SERIAL INTERFACE 37
4.14 RS485 SERIAL INTERFACE 38
4.15 VOLTAGE FREE RELAYS 38
4.16 ANALOGUE CONTACT INPUTS (ACI) 39

4.17 EARTH AND LIGHTNING PROTECTION 39
4.18 LED INDICATORS 40
4.19 NETFUSION-EXP SPECIFIC INTERFACES 40
4.20 CAN DUAL TRANSCEIVER 40
4.21 DUAL HIROSE EXPANSION DAUGHTER CARD SLOT 40

5 EXPANSION I/O DAUGHTER-CARD 42

5.1 DAUGHTER CARD DESIGN 43
5.2 GPIO HIROSE CONNECTORS 43
5.3 POWER SUPPLY 43
5.4 POWER TEST 44
5.5 IDC INTERFACES 44
5.6 EARTH PROTECTION 44
5.7 DAUGHTER-CARD EXPANSION I/O CONNECTORS 44
5.7.1 Connector Summary 44
5.7.2 Customer IDC (Outputs ONLY) Connector (J4) 45
5.7.3 Customer IDC I/O Connector (J5) 46
5.7.4 Customer IDC I/O Connector (J6) 47
5.7.5 Customer IDC (Inputs ONLY) Connector (J7) 48

6 HARDWARE DEVICE DRIVER SUPPORT 49

7 SQLITE USAGE 50

8 NETFUSION-EXP DEVELOPMENT-KIT WEB-SERVER 51

9 DEMO APPLICATION 53

CAT5E ETHERNET CABLE & CONNECTORS 54

9.1 RJ45 PLUG AND CAT5|6E CABLE CONNECTIONS 54

10 SMARTFUSION2 FPGA NETFUSION-EXP FABRIC DEVELOPMENT 56

10.1 LIBERO SOC/IDE VERSION 11.X DERIVATIVES 56
10.2 INSTALLATION 57
10.3 STARTING A PROJECT AND BASIC UNDERSTANDING 57
10.3.1 Libero 11.X IDE First Launch 58
10.4 NETFUSION-EXP 58
10.5 PRODUCT DEVELOPMENT 59

11 DOWNLOADING NEW IP CORE(S) INTO THE LIBERO IDE INSTALLATION 60

12 NETFUSION-EXP LIBERO I/O ASSIGNMENTS 61

13 NETFUSION-EXP LIBERO 11.X IDE/SOC TOP-LEVEL 63

13.1 EMCRAFT SYSTEMS 65
13.2 ULPI/UTMI OTG USB 66
13.3 CORESF2RESET 67
13.4 FCCC 68
13.5 COUNTER - COREGPIO 70

13.6 NETFUSION-EXP BASEBOARD PERIPHERAL COREGPIO 71
13.7 NETFUSION-EXP IO-EXP DAUGHTER CARD GPIO 72
13.8 BIBUF BI-DIRECTION BUFFERS FOR DAUGHTER CARD GPIO 73
13.9 CAN CONTROLLER 74
13.10 COREUARTAPB 75
13.11 CORESPI0, 1 76
13.12 TEMPERATURE SENSOR GLUE LOGIC 77
13.13 COORDINATED RESET 78
13.14 BIBUF 79
13.15 CORESF2CONFIG 80
13.16 CORE AXI 81
13.17 CORE AHB-LITE 82
13.18 CORE APB3 83
13.19 AXI, AHB-LITE & APB3 DEFAULT MEMORY MAPPED SETTINGS 84
13.20 SMARTFUSION2 MSS 87

14 NETFUSION-EXP ARM CORTEX MSS DEFAULT CONFIGURATION 88

14.1 CAN CONTROLLER 88
14.2 MDDR 89
14.3 USB 90
14.4 ETHERNET 91
14.5 MSS_CCC 92
14.6 RESET CONTROLLER 92
14.7 FIC_0 93
14.8 FIC_1 94
14.9 FIC_2 (PERIPHERAL INITIALIZATION) 95

15 ADDING IP CORES FROM VERILOG/VHDL 96

15.1 IMPORTING SOURCE FILES 96
15.2 INSTANTIATING INTO THE FABRIC 97
15.2.1 Building and Synthesizing the NetFusion-EXP Design 98

16 USING FLASHPRO4 IDE TO PROGRAM NETFUSION-EXP 100

17 FAQ 103

18 ENVIRONMENTAL SPECIFICATIONS 105

19 CONTACT 106

20 DOCUMENT HISTORY 107

List of Figures

Figure 1 - Power Connector Options 15
Figure 2 - Desk Mains Power Connected to Receptacle (Q2) 16
Figure 3 - Screw Terminal Power 17
Figure 4 - Microsemi FlashPro4 USB JTAG Programming 18
Figure 5 - USB Cable Connected to the USB-UART Interface from a PC 19
Figure 6 – Example of opening PuTTY in Windows 7 20
Figure 7 – An example on Ubuntu Linux of the Terminal being launched 20
Figure 8 – Screen shot of a User entering into the u-boot screen on the USB-UART terminal session 20
Figure 9 – Screen shot of allowing the boot process to execute uCLinux and initialize 21
Figure 10 - Block Diagram Architecture of the NetFusion-EXP main board PCB 27
Figure 11 – Inner FPGA Functional Block Diagram 28
Figure 12 - USB UART PC Typical Settings Configuration 29
Figure 13 - An Example of a terminal screen connected through USB-UART to Cortex-M3 uCLinux 30
Figure 14 - ARM JTAG Emulator & Debugger 32
Figure 15 - Other Examples of M3 JTAG Emulation Hardware 33
Figure 16 - A Standard typical 18V DC Desk Mains Supply 34
Figure 17 - 3.5mm Phono Plug (3-pin) 35
Figure 18 – Expansion Slot Area 41
Figure 19 - Main opening screen for the NetFusion-EXP Demo Website 51
Figure 20 – Drivers/Hardware Page for the NetFusion-EXP Demo Website 52
Figure 21 - Libero 11.X IDE Boot-Up Screen 58
Figure 22 - Libero Updating IP Core in the Vault 60
Figure 23 - Example of Top List I/O Ball Assignments 61
Figure 24 - Example of Bottom List I/O Ball Assignments 62
Figure 25 - Selecting the I/O Constraints 62
Figure 26 – NetFusion-EXP Top-Level Sheet 63
Figure 27 - Lower-Level SOM sheet MSS block 64
Figure 28 - Screenshot of Lower-Level SOM sheet of the Starter-Pack Libero Project 65
Figure 29 - ULPI/UTMI USB Converter IP Core 66
Figure 30 - SmartFusion2 Reset Controller IP Core 67
Figure 31 - Clock PLL Macro Core 68
Figure 32 - PLL Clock Macro Settings 69
Figure 33 - Temperature Sensor Logic 70
Figure 34 - Main PCB GPIO Interface with APB bus from the ARM Processor MSS block 71
Figure 35 – Main CoreGPIO for the Daughter Card Expansion IO slots 72
Figure 36 – Bi-Directional Buffers for the Daughter Card Expansion IO slots 73
Figure 37 - CAN Controller Interface Logic 74
Figure 38 - RS485 UART IP core 75
Figure 39 - Stereo Audio Line IN & Line OUT 76
Figure 40 - Temp Sensor Counter Logic 77
Figure 41- Customized RESET logic 78
Figure 42 - Bi-Directional Fabric Macros for input/output signals 79
Figure 43 - APB feedback Bus for Peripheral Configuration by Software 80
Figure 44 - AMBA Memory Interfaces from ARM for an AXI bus 81
Figure 45 - AMBA Memory Interfaces from ARM for an AHB-Lite bus 82

Figure 46 - APB3 fixed bus UART, GPIO, IO-EXP slots and Counter Peripherals 83
Figure 47 – Editing the APB3 core settings 83
Figure 48 - Default AXI Configuration 84
Figure 49 - Default AHB-Lite Configuration 85
Figure 50 - Default APB Configuration 86
Figure 51 - SOM Module of the ARM MSS 87
Figure 52 - Exploded View of the Modules in the NetFusion-EXP ARM MSS 88
Figure 53 - MDDR MSS Controlling LPDDR Memory with AXI from Fabric 89
Figure 54 - USB OTG UTMI Host Controller 90
Figure 55 - MSS MAC for 10/100 Ethernet Using SOM PHY 91
Figure 56 - MSS CCC Divider from the CLK_BASE with-in ARM Sub-System 92
Figure 57 - Reset MSS Module 92
Figure 58 - AHB/APB Fabric Interface [1] 93
Figure 59 - APB3/AHB-Lite Fabric Interface [2] 94
Figure 60 - APB Peripheral Hardware Configuration 95
Figure 61 - Importing a single Verilog/VHDL file into the NetFusion-EXP Libero Project 96
Figure 62 - Instantiating an Imported IP Core in your Fabric Design 97
Figure 63 - Selecting Full NetFusion-EXP Synthesis and FPGA Programming 99
Figure 64 - Opening screen of FlashPro IDE 100
Figure 65 - Microsemi FlashPro4 ribbon connection 101
Figure 66 - FlashPro Programming Screen 101

List of Tables

Table 1 : Voltage and Current Consumption on Power Supply 16
Table 2: RS485 Distance capabilities with respect to BAUD rate 38
Table 3: Summary of PCB Connectors 44
Table 4: IDC Connector J4 (Customer Interface) 45
Table 5: IDC Connector J5 (Customer Interface) 46
Table 6: IDC Connector J6 (Customer Interface) 47
Table 7: IDC Connector J7 (Customer Interface) 48
Table 8 : Full NetFusion User Space Hardware Drivers 49
Table 9 : Typical CAT5|6 cable wiring 55
Table 10: Document History Entry Log 107

About This Document

This specification fully documents and describes the NetFusion-EXP PCB incorporated as a Development-Kit package for R&D purposes. Use this manual to fully acquire a good understanding of the significant difference of the Development-Kit over and above the normal NetFusion-EXP delivery.

Important: *this document although a **Reference Guide PDF**, it does contain a section that warrants **User-Guide** material regarding quick setup and operational usage when interacting with the Development-Kit from your PC*.

Intended Audience

This document is fully written for Nine Ways' customers using the hardware NetFusion-EXP Development-Kit for evaluation and R&D.

1 Introduction

The NetFusion-EXP Ethernet SWITCH Development-Kit is an added layer of deliverable package to the customer. Whereas the ordinary NetFusion-EXP product delivery is a packaged baseboard PCB, whether it is for evaluation, test or full industrial deployment and mass-production purposes, the Development-Kit EXP bundles a whole item list of additional equipment.

Tailored for the Research & Development process, if you choose to order and use the Development-Kit-EXP, you will be provided and equipped with *cabling, USB cables, LCD screen, desk pillar support, Ethernet CAT5/6e patch cable, audio phone stereo patch cables, RS232 modem cable and an L.T.E 19V 120W desk mains brick*. These are all additional items over the standard NetFusion-EXP PCB deliverable that you will receive as part of the Development-Kit package.

The core of the architecture is the **S**ystem-**O**n-**M**odule (***SOM***) M2S050-SOM-F484 is the Smartfusion2 FPGA and ARM Cortex-M3 sub-system. The Flash, RAM, clocks, default Ethernet PHY, FPGA and all essential voltage level generation are contained on the SOM plug-in to the NetFusion-EXP. All the connections from the SOM to the NetFusion-EXP baseboard go through durable connectors and allow for inter-changeability if needed or required by the user.

The usage of a SOM on the baseboard reduces the risk of writing-off the value of the sub-system when interfacing to the real-world, damage occurs or installation faults. As a baseboard, it is ideal for a user to take advantage of the Smartfusion2 FPGA technology whilst at the same time, binding it with the industrial hardware interfaces and environmental protection on the connectors and over engineered power supply system.

NetFusions' are sold as variants of a core family – and the EXP variant concentrates on the hardware supporting *10/100* Ethernet with a wide ranging functional daughter card expansion capability..

Obviously, the PCB hosts a range of supporting hardware for customer interfacing and communication to allow the product to have a usage and functional reason to be deployed by the user.

2 Established Documentation

2.1 Basic NetFusion Concept, Benefits and Brief

Please download and refer to the **NetFusion Baseboard Product Brief PDF** at the Nine Ways R&D website.

2.2 Libero SmartFusion2 FPGA Fabric

The NetFusion-EXP Development-Kit promotes the use of a free *Libero IDE* project to download into the baseboard PCB via JTAG.

2.3 Hardware PCB Reference

Please download and refer to the **NetFusion-EXP Hardware Reference Guide PDF** at the Nine Ways R&D website.

2.4 Full Software & uCLinux Support and Operational Support

Please download and refer to the **NetFusion Software uCLinux User Guide PDF** at the Nine Ways R&D website.

2.5 Network Performance and Speed Testing

Please download and refer to the **DMA 'iperf' Test App User Guide PDF** at the Nine Ways R&D website.

Note: *You may be slightly confused by some of the documented illustrations and instructions as this is written to test on an Emcraft Development-Kit board. However, the principles are the same and the instructions on how to launch and test end-to-end connectivity do not differ.*

2.6 Relevant References

As additional help and support of the Software and the uCLinux environment please consider the following resources:

Visit Emcraft's web site at www.emcraft.com to obtain additional materials related to Linux Cortex-M.

Emcraft Systems provides detailed kit information and materials from this page:
http://www.emcraft.com/som/m2s

As new SmartFusion2 and kit related materials become available, they will be posted on the Microsemi and Emcraft Systems web sites. Microsemi recommends that you sign up for Product Updates to be notified when new material is available.

Microsemi product updates: https://www.actel.com/portal/default.aspx?r=2

Emcraft Systems product updates: http://www.emcraft.com/som/m2s/register

Refer to *Emcraft Systems SmartFusion2 SOM (System-On-Module) Hardware Architecture* for detailed information on the hardware architecture of the Emcraft Systems SmartFusion2 SOM board.

Refer to *Emcraft Systems SOM-BSB-EXT Baseboard Hardware Architecture* for detailed information on the hardware architecture of the Emcraft Systems SOM-BSB-EXT baseboard.

Refer to *Linux Cortex-M User's Manual* for detailed information on the software architecture of the Linux SmartFusion2 distribution.

For more of the interested readers on the hardware resources for the NetFusion-EXP PCB:

Robust Electronic Design Reference Book, Volumes I and II.

Surge Protection Reference Guide - Emerson Network Power.

Earthing guide for surge protection - MTL Instruments Group.

What's The Difference Between The RS-232 And RS-485...

GMII Timing and Electrical Specification.

Design Resources | FPGA & SoC | Products - Microsemi.

Engineer's guide to High-Quality PCB Design document.

3 Out of the Box, Development-Kit, Powering Up & Programming

The product should be presented boxed, bubble wrapped and protected with anti-static packaging. Be careful not to static discharge yourself through the PCB as this can cause severe if not irreparable damage.

Note: *always try to use earth straps when handling the NetFusion-EXP PCB.*

3.1 Kit Contents

1 x NetFusion-EXP PCB with LCD, nuts, bolts, screws, washers + 4 pillar legs for desk usage

1 x RJ45 Cat5/6e Ethernet patch lead

1 x USB USB-UART lead for using with PuTTY communicating with a terminal session

2 x Stereo phono-jack to individual socket phono cables (this is often supplied as metal plugs for the user to solder to with their own cabling – shown on the right above)

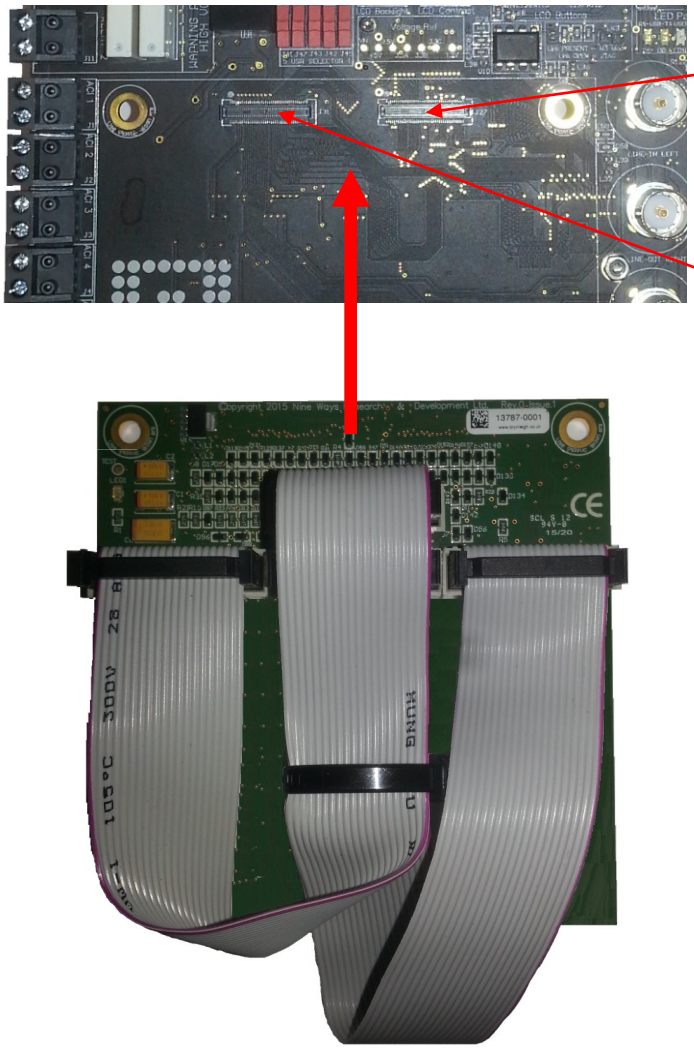
1 x RS232 Straight modem cable

*1 x 19V 120W desk mains supply (*110/230V 50/60Hz*)*

Depending on the country of order, the mains plug will be one of the following:

1 x British Plug cable **OR**

1 x Euro Plug cable **OR** *1 x USA Standard Plug cable*



Slot on the EXP-Daughter Card I/O PCB. This just utilizes the top space on the overall much larger expansion area on the NetFusion-Exp base-board. You do not have to worry about fixing into the screw holes for this test either.

1 x Expander I/O PCB (the Dev-Kit always has this even if the user wishes to use their own designed plug-in board eventually).

Note: *The ribbon-cable shown are not included .*

3.2 Applying Power

The product should be presented boxed, bubble wrapped and protected with anti-static packaging. Be careful not to static discharge yourself through the PCB as this can cause severe if not irreparable damage.

Note: *always try to use earth straps when handling the NetFusion-EXP PCB.*

3.3 Applying Power

There are two different methods of supplying the DC power to the NetFusion-EXP product. They are electrically wired in parallel and should not be used simultaneously, even though there is on-board protection for such a circumstance.

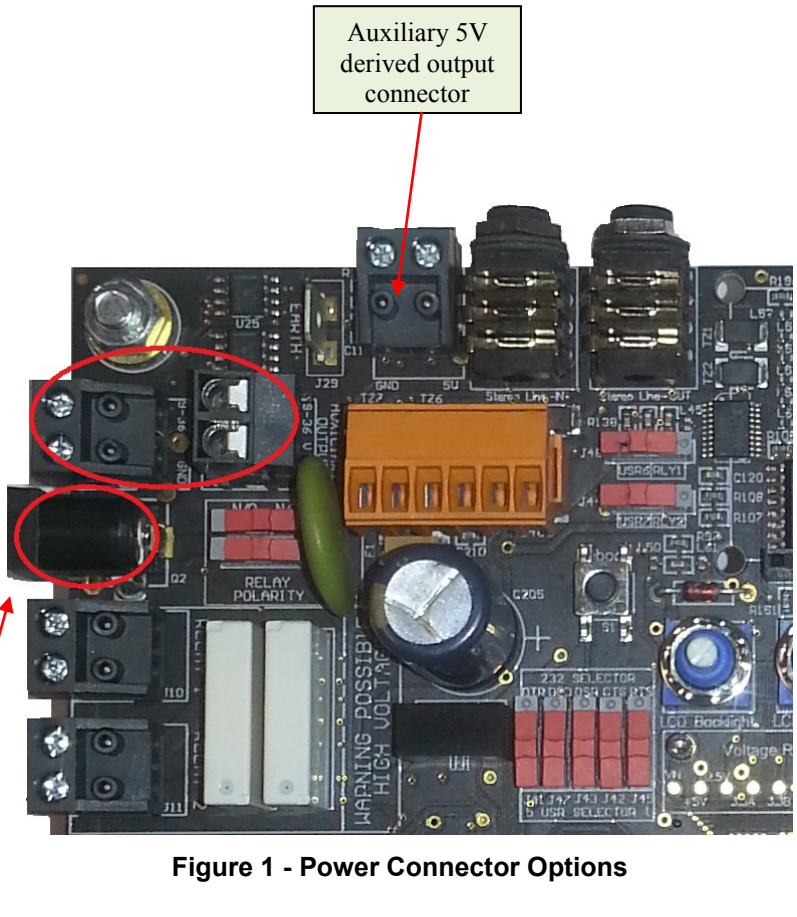


Figure 1 - Power Connector Options

Both power methods allow for 9-36V DC voltage variation.

Important Note: *you should always consider equipping and providing a power supply delivery unit that provides at least twice the maximum current consumption in the table below.*

Important Note: *The below table assumes a current consumption of an average of 500mA constant from a daughter card expansion.*

Input Voltage	Current Consumption @ Maximum FPGA Fabric Usage and Switching
9	1920mA
12	1390mA
18	1130mA
24	850mA
36	660mA

Table 1 : Voltage and Current Consumption on Power Supply

3.3.1 Bench Supply

You can use a standard **18V L.T.E**. or similar mains desk supply. The molded connector on the mains supply will plug directly into the power receptacle (**Q2**) on the NetFusion-EXP PCB.

Figure 2 - Desk Mains Power Connected to Receptacle (Q2)

This method of connection is ideal for desk, evaluation, development and test. It avoids the cumbersome bench supply and allows for product usage when there is no industrial supply. If you use NetFusion-EXP as a *Development Kit* for instance, this supply will be provided with the delivery.

3.3.2 Screw Terminal Installation

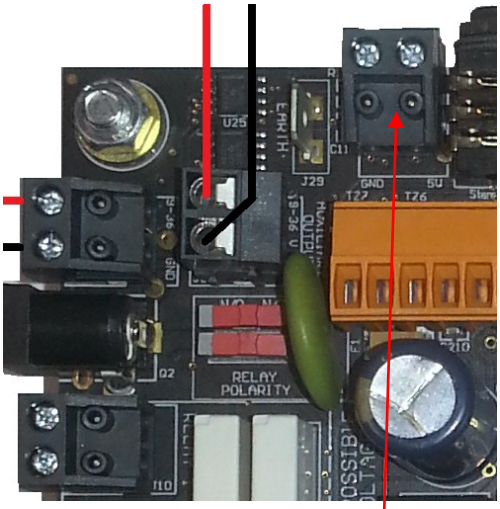


Figure 3 - Screw Terminal Power

Whilst only one of the two pairs of power is necessary, having a dual approach allows for daisy chaining equipment, or having two parallel bench or industrial power available *(if current is limited to less than 500mA per output unit*).

3.3.3 Output 5V Auxiliary

Next to the two screw-terminal power supply connectors is a +5V DC derived output supply. This is generated on-board NetFusion-EXP and can be used to supply external *TTL/CMOS* peripheral devices connected to the baseboard. This may not be necessary when powering up the board for the first time or getting started, but it is mentioned so that you do not get this confused with the input DC power supply connectors.

Please Note: *maximum current delivery capability of the auxiliary supply is 350mA @ 5V.*

3.3.4 JTAG Programming

With the on-board PCB *JTAG programming mode LINK* removed and not present, the SmartFusion2 FPGA on the SOM will accept normal JTAG programming, validation and verification. This is all controlled using the FlashPro IDE from Microsemi or the FlashPro embedded in Libero SoC/IDE.

Figure 4 - Microsemi FlashPro4 USB JTAG Programming

Firstly, check that the two Red LED power indicators are illuminated on the NetFusion-EXP PCB. With the JTAG programming mode LINK removed, plug in the USB programmer (*as shown above*). Using the Microsemi IDE application, program your IP fabric and eNVM u-boot image.

Note: *for Libero IDE, FlashPro4 & uCLinux compilation, please see the other NetFusion-EXP PDF guides.*

Once the Microsemi IDE has signaled that the programming operation was successful and completed, you can optionally remove the JTAG ribbon cable from the header on the PCB. You **must** remove and then re-apply the DC power.

Once the power is restored and the NetFusion-EXP System-On-Module's SmartFusion2 FPGA is operating and the Cortex-M3 is launching the eNVM RAM u-boot, progress to the following sub-section (*below*).

3.3.5 USB UART Terminal

As the SOM's Cortex-M3 processor (*embedded inside the SmartFusion2 FPGA*) begins to boot-up, the debug and process can be observed using the USB-UART interface. This is a USB device that your connected PC will recognize as a USB Serial Device and allocate a COM port to it (Windows XP/7/8)
Note: */dev/ttyS0|1|2 for Linux PCs.*

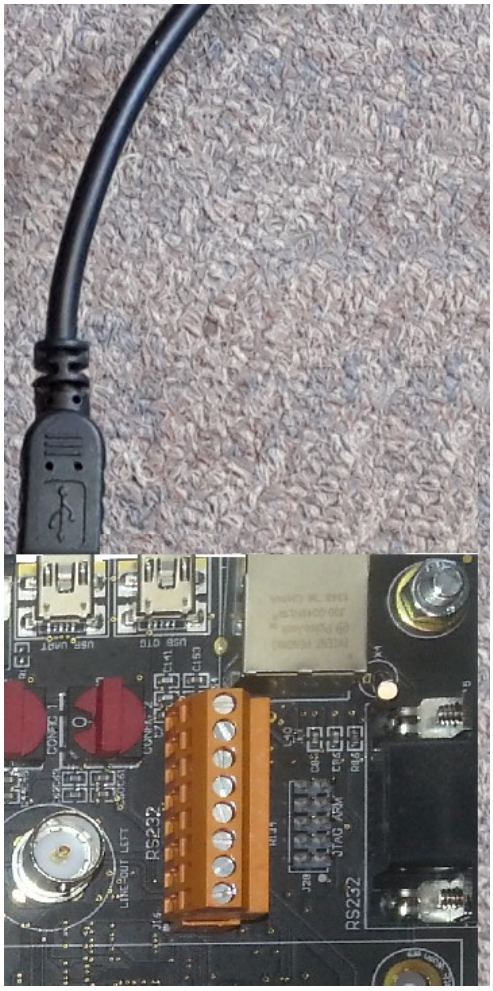


Figure 5 - USB Cable Connected to the USB-UART Interface from a PC

The IDE on the PC should open a window that can see the debug and allow for commands and text to be entered. This will interact with the u-boot.

Once you have programmed the SPI Flash with the uCLinux (*see* NetFusion uCLinux Guide PDF) then the interaction and debug will be with the booted uCLinux. This includes how to connect a CAT5 to the *10/100* RJ45 for *TFTP* network transfer of the Linux kernel file.

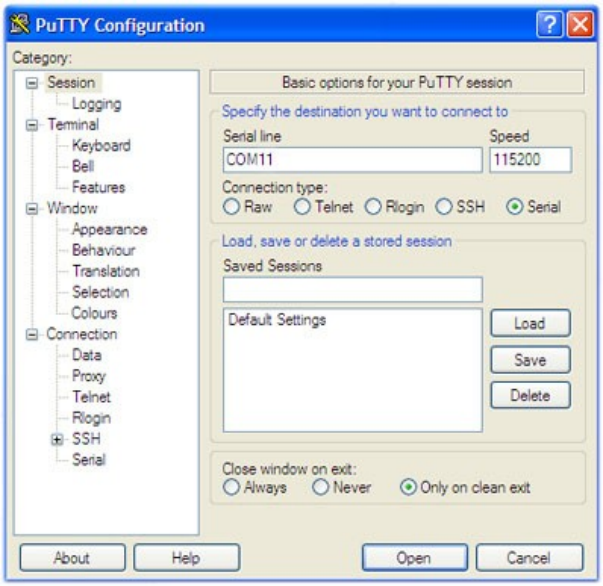


Figure 6 – Example of opening PuTTY in Windows 7

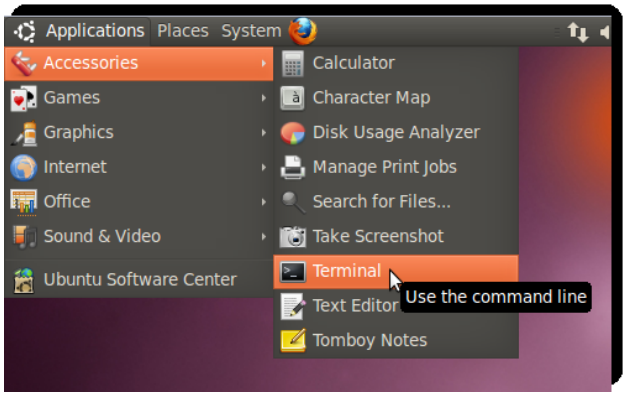


Figure 7 – An example on Ubuntu Linux of the Terminal being launched

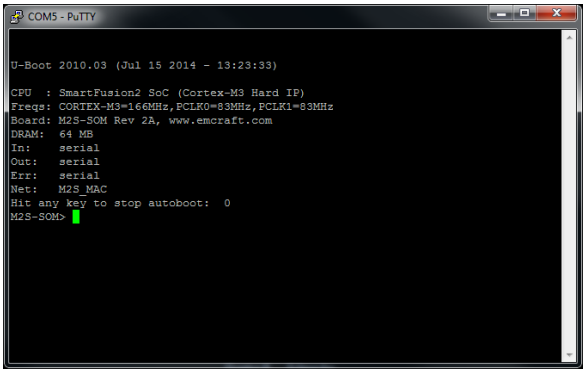


Figure 8 – Screen shot of a User entering into the u-boot screen on the USB-UART terminal session

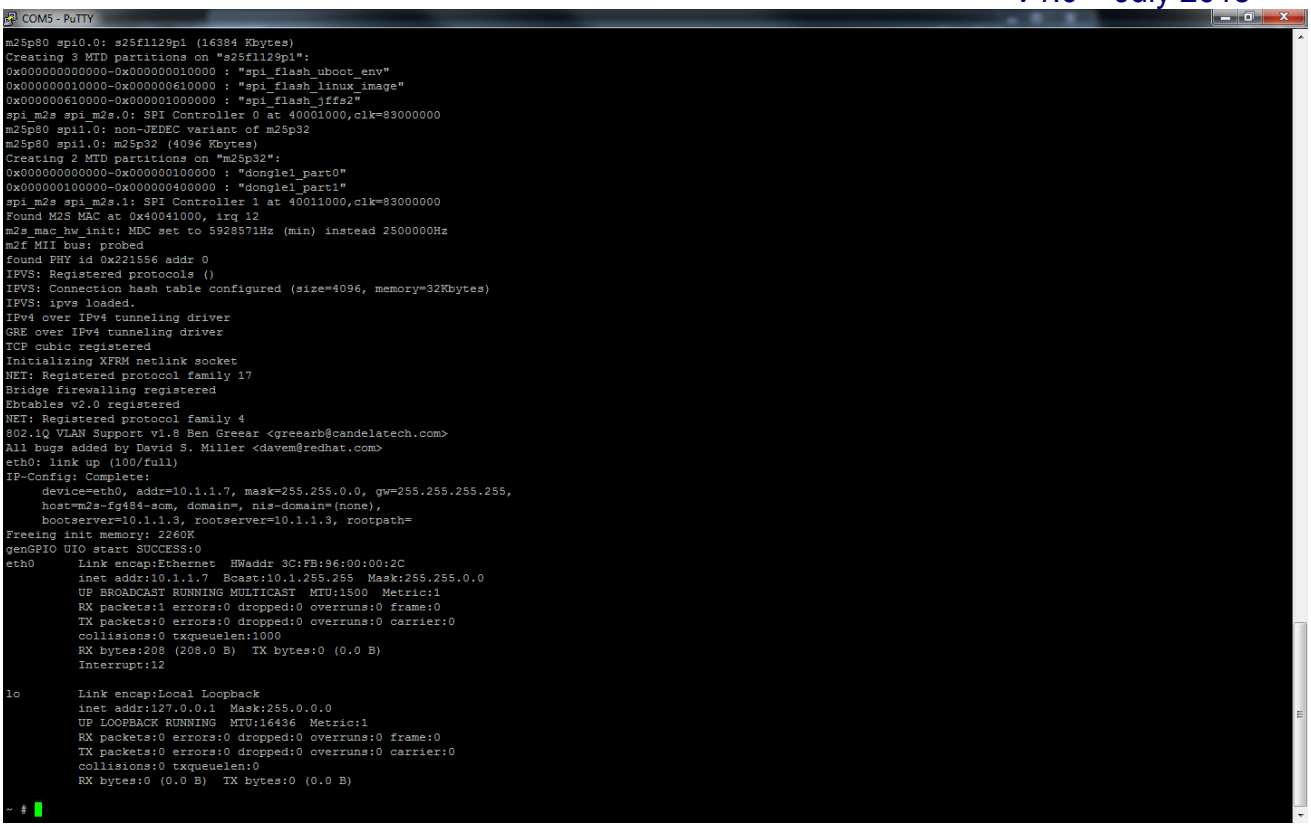

COM5 - PuTTY
m25p80 spi0.0: s25fl129p1 (16384 Kbytes)
Creating 3 MTD partitions on "s25fl129p1":
0x000000000000-0x000000010000 : "spi_flash_uboot_env"
0x000000010000-0x000000610000 : "spi_flash_linux_image"
0x000000610000-0x000001000000 : "spi_flash_jffs2"
spi_m2s spi_m2s.0: SPI Controller 0 at 40001000,clk=83000000
m25p80 spi1.0: non-JEDEC variant of m25p32
m25p80 spi1.0: m25p32 (4096 Kbytes)
Creating 2 MTD partitions on "m25p32":
0x000000000000-0x000000100000 : "dongle1_part0"
0x000000100000-0x000000400000 : "dongle1_part1"
spi_m2s spi_m2s.1: SPI Controller 1 at 40011000,clk=83000000
Found M2S MAC at 0x40041000, irq 12
m2s_mac_hw_init: MDC set to 5928571Hz (min) instead 2500000Hz
m2f MII bus: probed
found PHY id 0x221556 addr 0
IPVS: Registered protocols ()
IPVS: Connection hash table configured (size=4096, memory=32Kbytes)
IPVS: ipvs loaded.
IPv4 over IPv4 tunneling driver
GRE over IPv4 tunneling driver
TCP cubic registered
Initializing XFRM netlink socket
NET: Registered protocol family 17
Bridge firewalling registered
Ebtables v2.0 registered
NET: Registered protocol family 4
802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
All bugs added by David S. Miller <davem@redhat.com>
eth0: link up (100/full)
IP-Config: Complete:
device=eth0, addr=10.1.1.7, mask=255.255.0.0, gw=255.255.255.255,
host=m2s-fg484-som, domain=, nis-domain=(none),
bootserver=10.1.1.3, rootserver=10.1.1.3, rootpath=
Freeing init memory: 2260K
genGPIO UIO start SUCCESS:0
eth0 Link encap:Ethernet HWaddr 3C:FB:96:00:00:2C
inet addr:10.1.1.7 Bcast:10.1.255.255 Mask:255.255.0.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:208 (208.0 B) TX bytes:0 (0.0 B)
Interrupt:12
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
~ #

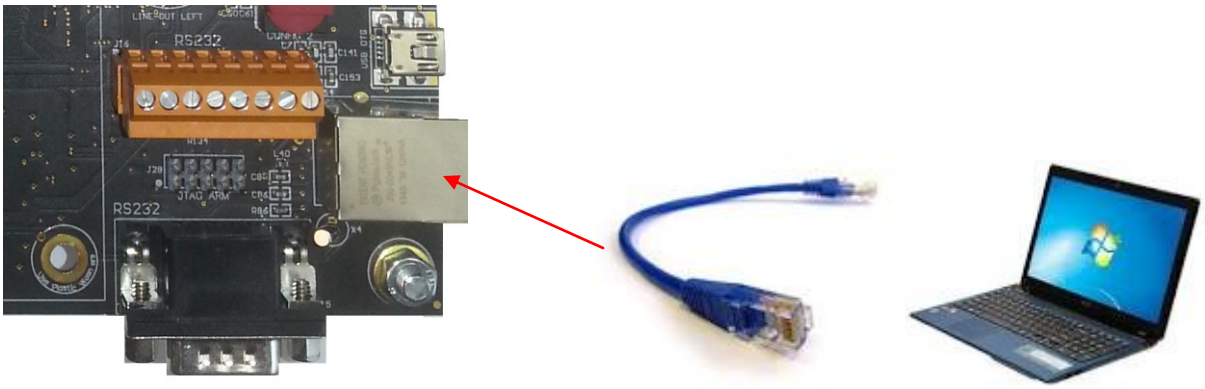
Figure 9 – Screen shot of allowing the boot process to execute uCLinux and initialize

Your *IDE* on the PC should open a window that can see the debug and allow for commands and text to be entered. This will interact with the u-boot and once you have programmed the SPI Flash with the uCLinux (*see* ***NetFusion Software uCLinux Guide PDF***) then the interaction and debug will be with the booted uCLinux. This includes how to connect a CAT5 to the *10/100 RJ45* for *TFTP* network transfer of the Linux kernel file.

3.4 Connecting a CAT5 Cable for TFTP Upload

This document specifically does not go into detail on the installation, cross-compiling of the embedded code, but below in simplistic ways illustrates the steps to upload the compiled and built uCLinux kernel and software:

Place a CAT5 cable (*provided*) between your Laptop PC's Ethernet port and the NetFusion-EXP Ethernet port:



3.5 Building the NetFusion Customized uCLinux Kernel

The entire NetFusion uCLinux environment can be built from clean in two simple steps. For a normal incremental make which just updates any changes, then it can be generally built in a single step.

Important Note*: if you have just booted your PC, or have closed down a terminal session and then re-started, you will have to go through an initialization step first.*

You must activate your uCLinux Cortex-M development session by navigating to the top of your uCLinux Cortex-M installation and run:

```
[psl@pvr linux-cortexm-1.12.0]$ source ACTIVATE.sh
```

Then change into the *netfusion/* directory

```
linuxPC:/ Project/linux-cortexm-1.xx.x/ # cd projects/netfusion
```

Also, if you had to run the *source ACTIVATE.sh* (*above*), then you must also run the following:

```
linuxPC:/ Project/linux-cortexm-1.xx.x/ projects/netfusion # ./startftftp
```

To clean the entire project type:

```
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion # make clean
```

Then to clean the SQLite separate build system type:

```
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion # cd source/sqlite
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/sqlite # make clean
```

Then to clean the demo application separate build system type:

```
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion source/sqlite # cd ..
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source # cd demoapp
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/demoapp # make clean
```

Then to clean the linuxptp separate build system type:

```
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/demoapp # cd ..
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source # cd linuxptp
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/linuxptp # make clean
```

Then to clean the 'iperf' test application separate build system type:

```
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/linuxptp # cd ..
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source # cd testapp
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/testapp # make clean
```

Then make the full uCLinux build with NetFusion (*including the SQLite3, linuxptp, testapp & demo application binaries*) by typing:

```
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/testapp # cd ..
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source # cd ..
liuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion # make –f all
```

Alternatively make just the ./source directory files only type:

```
liuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion # make
```

Finally, prepare the newly generated uCLinux kernel package binary for TFTP upload:

```
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/ # ./savetftp
```

3.5.1 Building the SQLite, linuxptp, testapp & demo application incrementally

Change into the *netfusion/source/sqlite* directory and build the binary with *make*

```
linuxPC:/ Project/linux-cortexm-1.xx.x/ # cd projects/netfusion/source/sqlite
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/sqlite # make
```

Change into the *linuxptp* directory and build the binary with *make*

```
linuxPC:/ Project/linux-cortexm-1.xx.x/ # cd ../linuxptp
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/linuxptp # make
```

Change into the *demoapp* directory and build the binary with *make*

```
linuxPC:/ Project/linux-cortexm-1.xx.x/ # cd ../demoapp
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/demoapp # make
```

Change into the *testapp* directory and build the binary with *make*

```
linuxPC:/ Project/linux-cortexm-1.xx.x/ # cd ../testapp
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/source/testapp # make
```

3.5.2 Building the main uCLinux with NetFusion Incrementally

Then change into the *netfusion/* directory and build the binary with *make*

```
linuxPC:/ Project/linux-cortexm-1.xx.x/ # cd projects/netfusion
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion # make
```

Finally, prepare the newly generated uCLinux kernel package binary for TFTP upload:

```
linuxPC:/ Project/linux-cortexm-1.xx.x/projects/netfusion/ # ./savetftp
```

3.5.3 Uploading the kernel to the NetFusion

To load the image directly, use the *netboot* u-boot macro:

```
M2S-SOM> setenv image netfusion.uImage
M2S-SOM> run netboot
...
TFTP from server 172.17.0.1; our IP address is 172.17.5.100
Filename 'networking.uImage'.
...
Loading: #################################################################
         #################################################################
         #############
done
Bytes transferred = 2084704 (1fcf60 hex)
...
   Image Name:   Linux-2.6.33-arm1
   Image Type:   ARM Linux Kernel Image (uncompressed)
...
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
OK
Starting kernel ...

Linux version 2.6.33-arm1 (gcc version 4.4.1 (Sourcery G++ Lite 2010q1-189) ) #1 Fri Dec 06
15:43:44 MSK 2013
...
```

To load the image into the flash, use the u-boot `update` macro:

```
M2S-SOM> setenv image netfusion.uImage
M2S-SOM> run update
...
TFTP from server 172.17.0.1; our IP address is 172.17.5.100
Filename 'networking.uImage'.
...
Loading: #################################################################
         #################################################################
         #############
done
Bytes transferred = 2084704 (1fcf60 hex)
16384 KiB S25FL128S_64K at 0:0 is now current device
Saving Environment to SPI Flash...
Erasing SPI flash...Writing to SPI flash...done
M2S-SOM>
```

Reset the NetFusion baseboard and verify that the newly programmed image boots in the autoboot mode:

```
M2S-SOM> reset
resetting ...

U-boot 2010.03-linux-cortexm-1.12.0 (Dec 06 2013 - 17:19:37)
...
Starting kernel ...
...
init started: BusyBox v1.17.0 (Dec 06 2013 - 17:21:30)
~ #
```

3.5.1 Full uCLinux Documentation & Software Development

The (*above*) steps serve as a simple guide to try to successfully establish an uCLinux re-compilation, build and the upload via *TFTP* (*CAT5 network cable*) using u-boot. However, this topic and documented material is fully covered and explained with all pit-falls and extensive explanation in the ***NetFusion Software uCLinux Guide PDF*** available from the Nine Ways website.

4 NetFusion-EXP Hardware PCB

4.1.1 Net Fusion-EXP Baseboard Block Diagram

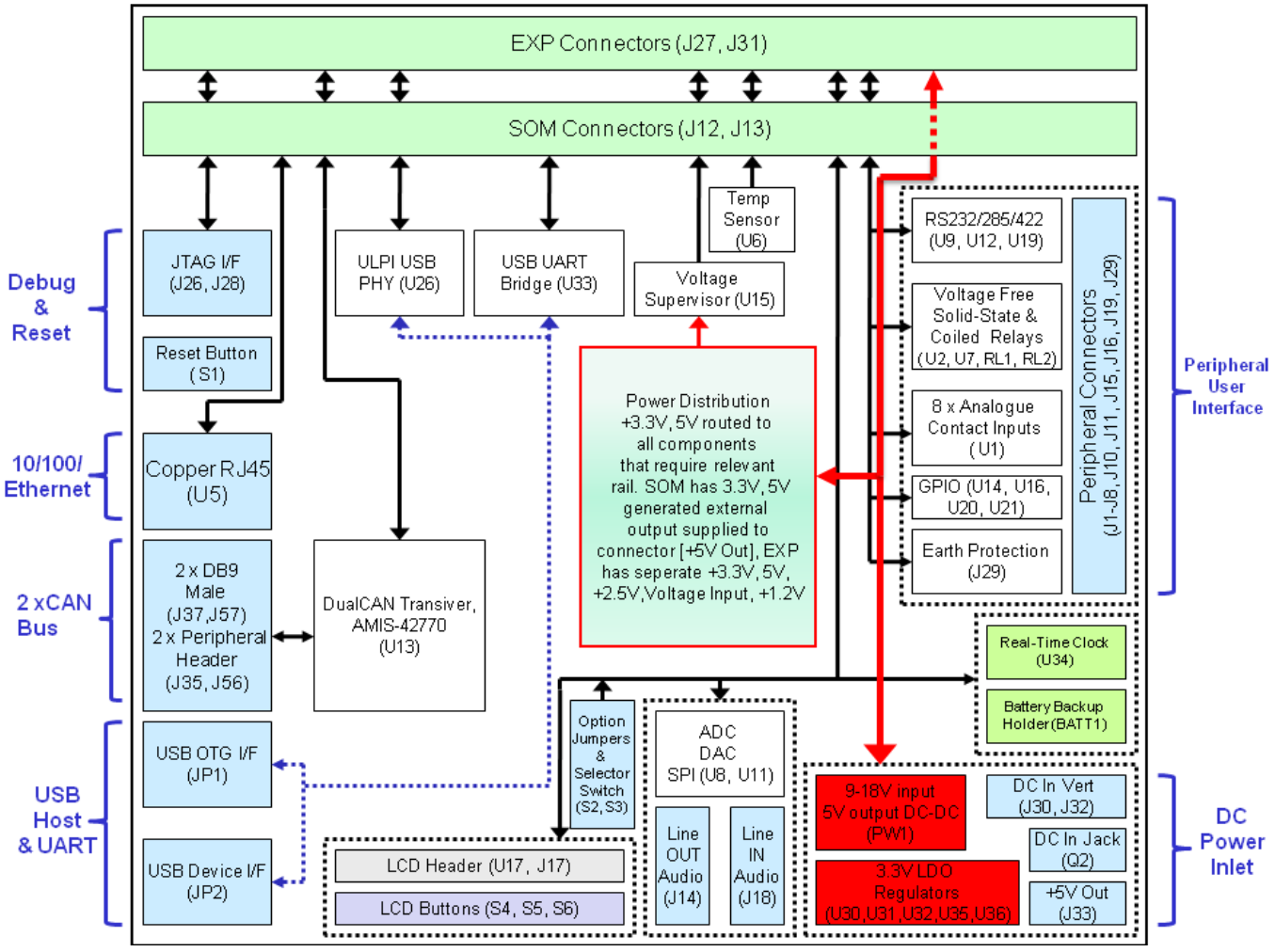

EXP Connectors (J27, J31)
SOM Connectors (J12, J13)
Debug & Reset
JTAG I/F (J26, J28)
Reset Button (S1)
10/100/ Ethernet
Copper RJ45 (U5)
2 xCAN Bus
2 x DB9 Male (J37,J57) 2 x Peripheral Header (J35, J56)
DualCAN Transiver, AMIS-42770 (U13)
USB Host & UART
USB OTG I/F (JP1)
USB Device I/F (JP2)
ULPI USB PHY (U26)
USB UART Bridge (U33)
Temp Sensor (U6)
Voltage Supervisor (U15)
Power Distribution +3.3V, 5V routed to all components that require relevant rail. SOM has 3.3V, 5V generated external output supplied to connector [+5V Out], EXP has seperate +3.3V, 5V, +2.5V, Voltage Input, +1.2V
Option Jumpers & Selector Switch (S2, S3)
LCD Header (U17, J17)
LCD Buttons (S4, S5, S6)
ADC DAC SPI (U8, U11)
Line OUT Audio (J14)
Line IN Audio (J18)
RS232/285/422 (U9, U12, U19)
Voltage Free Solid-State & Coiled Relays (U2, U7, RL1, RL2)
8 x Analogue Contact Inputs (U1)
GPIO (U14, U16, U20, U21)
Earth Protection (J29)
Peripheral Connectors (J1-J8, J10, J11, J15, J16, J19, J29)
Peripheral User Interface
Real-Time Clock (U34)
Battery Backup Holder (BATT1)
9-18V input 5V output DC-DC (PW1)
3.3V LDO Regulators (U30, U31, U32, U35, U36)
DC In Vert (J30, J32)
DC In Jack (Q2)
+5V Out (J33)
DC Power Inlet

Figure 10 - Block Diagram Architecture of the NetFusion-EXP main board PCB

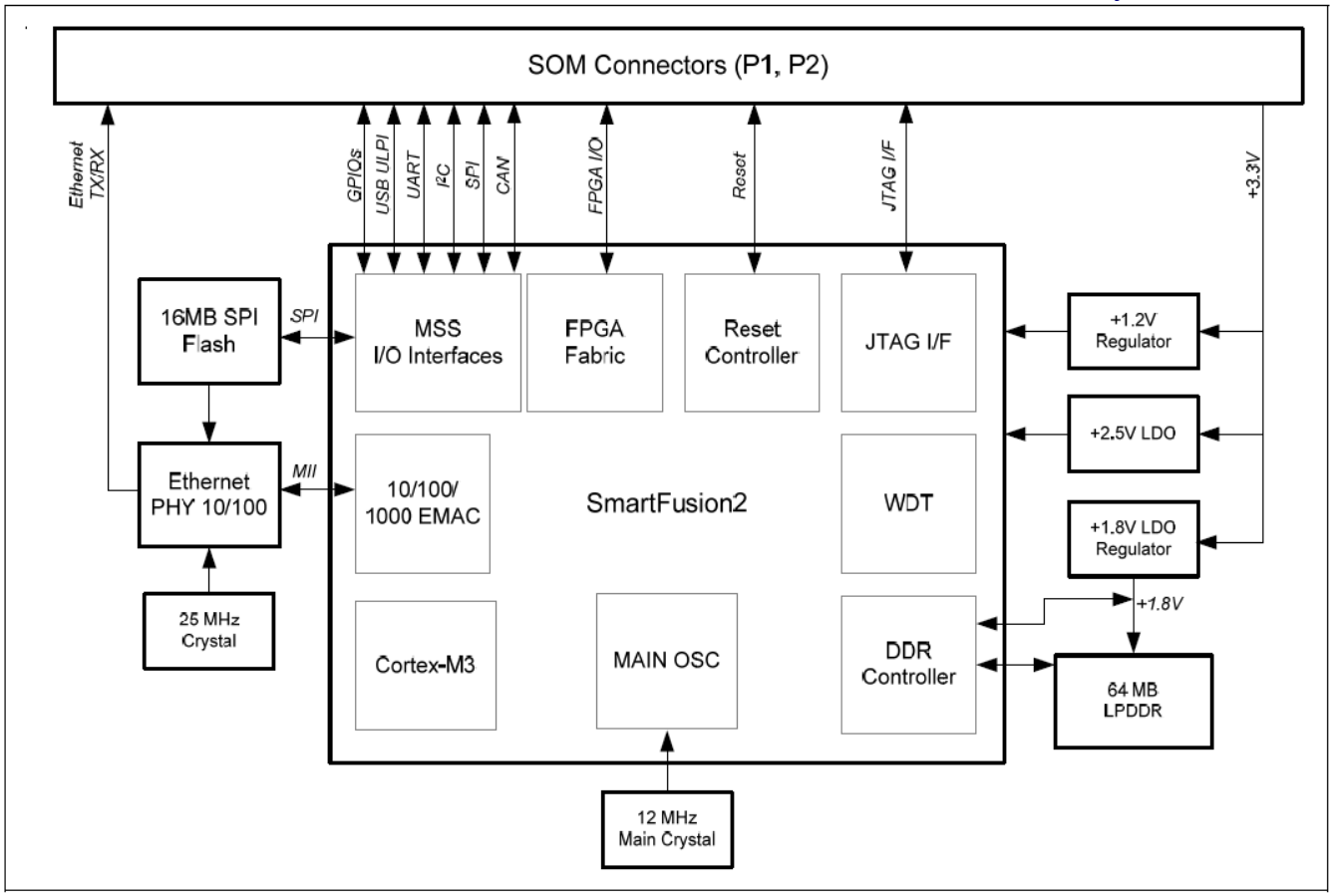


Figure 11 – Inner FPGA Functional Block Diagram

The NetFusion-EXP Development-Kit which is based on the NetFusion-EXP PCB baseboard has two layers of functionality in hardware. The first block diagram is the actual main PCB that houses the SOM. The SOM contains the FPGA and the surrounding hardware echo-system. This is shown in the diagram directly above.

Essentially, in both diagrams the SOM Headers are the same connector mated together.

4.2 Component Description Details

Before you read the sub-sections on the hardware elements available to the user, please refer to the **NetFusion-EXP Hardware Reference Guide PDF** available from the Nine Ways R&D website for all tables and pin-outs that you can cross-reference with the hardware listed and described (*below*).

4.2.1 USB UART Interface

The NetFusion-EXP PCB provides a USB UART interface on the type-B mini USB connector **(JP2).** The interface is provided using an *FTDI FT232RL USB-UART* bridge **(U33)**. The *FT232R*L UART TX and RX pins are connected to the SOM serial console UART port.

The main USB interface is a more default, statutory interface for the reason of commanding the NetFusion-EXP on loading code into the memory and viewing debug. It is the main viewing portal for the Linux OS inside the SOM, although a *user/customer* can during run-time change the usage of this port to suit their own needs.

The *FL232 SM IC* on the NetFusion PCB is a USB device that presents itself on a connected Windows/Linux/Mac PC as a USB Serial Device. This has a programmed baud rate and other attributes as you would logically expect but the default is *115200 8N1*.

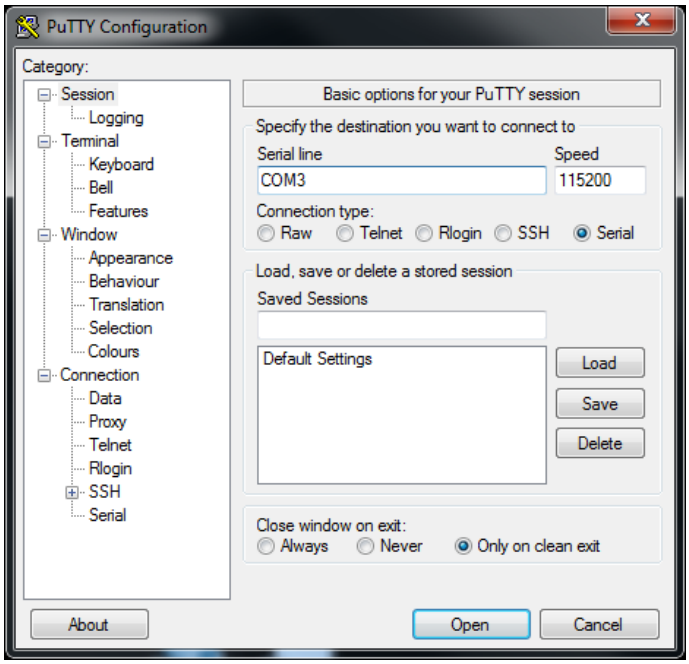


Figure 12 - USB UART PC Typical Settings Configuration

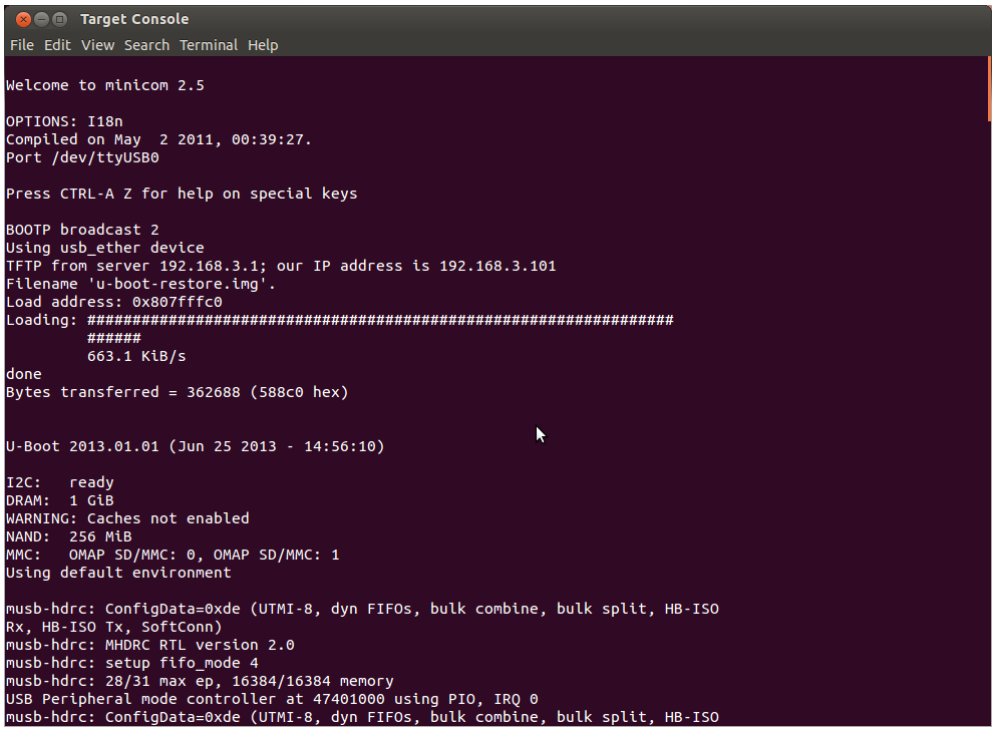

Target Console
File Edit View Search Terminal Help
Welcome to minicom 2.5
OPTIONS: I18n
Compiled on May 2 2011, 00:39:27.
Port /dev/ttyUSB0
Press CTRL-A Z for help on special keys
BOOTP broadcast 2
Using usb_ether device
TFTP from server 192.168.3.1; our IP address is 192.168.3.101
Filename 'u-boot-restore.img'.
Load address: 0x807fffc0
Loading: ##
######
663.1 KiB/s
done
Bytes transferred = 362688 (588c0 hex)
U-Boot 2013.01.01 (Jun 25 2013 - 14:56:10)
I2C: ready
DRAM: 1 GiB
WARNING: Caches not enabled
NAND: 256 MiB
MMC: OMAP SD/MMC: 0, OMAP SD/MMC: 1
Using default environment
musb-hdrc: ConfigData=0xde (UTMI-8, dyn FIFOs, bulk combine, bulk split, HB-ISO
Rx, HB-ISO Tx, SoftConn)
musb-hdrc: MHDRC RTL version 2.0
musb-hdrc: setup fifo_mode 4
musb-hdrc: 28/31 max ep, 16384/16384 memory
USB Peripheral mode controller at 47401000 using PIO, IRQ 0
musb-hdrc: ConfigData=0xde (UTMI-8, dyn FIFOs, bulk combine, bulk split, HB-ISO

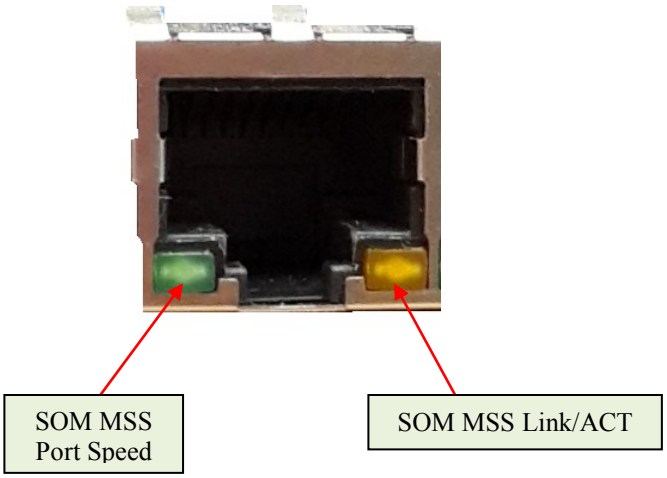
Figure 13 - An Example of a terminal screen connected through USB-UART to Cortex-M3 uCLinux

4.2.2 Ethernet Interface

The RJ45 Etherent port is wired directly through the SOM connectors to the SOM on-board PHY and dedicated MAC to the MSS for the ARM processor. The maximum speed of this port is *100Mbit/s*. This is due to the PHY on the SOM and the usage of only an MII 25MHz interface into the SmartFusion2 FPGA embedded MSS MAC. By default it is used as a means to allow re-compiled uCLinux kernels and application code to be programmed into the SPI FLASH memory using *TFTP*. The commands are initiated using the USB-UART interface (*above*). It also is then useful as the "default" */dev/eth0* in the Linux system and provides an immediate Ethernet interface without any added IP cores in the FPGA fabric.

LED Definitions:

Note: *For DUPLEX LEDs:* ***ON*** *= FULL* ***OFF*** *= HALF. For SPEED LEDs:* ***ON*** *= 100Mbit/s* ***OFF*** **=** *10Mbit/s*



4.1 USB OTG Interface

The second USB port on the NetFusion-EXP PCB baseboard has no default purpose. The traces from the USB host hardware on the PCB trace route through to the SOM SmartFusion2 FPGA fabric and present to a ULPI/UTMI converter IP core. This can be used either in the fabric or to the MSS ARM Cortex-M3 processor embedded MSS USB support.

A customer/user can choose how to use this port and has complete control of the usage and purpose. The OTG "*on-the-go*" mechanism is very flexible and is widely recommended in the USB industry.

Important Note: *the SmartFusion2-SOM-M2S-050 does not implement ULPI in the MSS USB peripheral so the fabric has the ULPI/UTMI converter for the user's convenience when they plan how to use the USB.*

For useful documents on understanding USB OTG then please refer to: OTG Specification - USB.org

4.2 FlashPro JTAG Interface

The JTAG header allows for the *10-way* ribbon cable from a FlashPro USB programmer to access the SmartFusion2 FPGA. In particular the actions relate to programming the eNVM Flash with the u-boot code to boot the device from reset. Moreover, it access and re-programs all of the FPGA fabric flash that houses the IP cores and project integration.

The programmer is controller on a host PC by *Libero IDE* or FlashPro4 stand-alone application. Just connect the ribbon from the programmer onto the header (*make sure that the RED ribbon line is on the same side as the PCB silkscreen dot*). The IDE handles all of the integration of the u-boot code and the fabric. This goes beyond the scope of the document from here.

Once the SmartFusion2 FPGA on the SOM is released from a full reset, the fabric flash is loaded into the SDRAM of the core and provides the RAM based fabric operation. Additionally, the eNVM flash is loaded into the eNVM RAM for u-boot execution. It is at this stage that the u-boot loads the uCLinux from the SPI external flash on the SOM into the external LPDDR SDRAM memory. The Linux environment then boots and execution of the user applications can begin.

4.3 ARM JTAG Interface

The NetFusion-EXP PCB provides an ARM JTAG interface for debugging of the SmartFusion2 SoC FPGA on the SmartFusion2 SOM. The connector is a standard *20-pin* JTAG port supported by standard Cortex-M3 programmers/emulators.

When using this function, make sure that the option jumper *LINK* setting is correct for the JTAG mode.

Figure 14 - ARM JTAG Emulator & Debugger

For specific information on the usage and the datasheet of the Cortex-M3 ARM Debugger please refer to the following document: Download JTAGJET-CORTEXM3 datasheet.

There are other manufacturers of similar applicable products that interface with the M3 Debugging JTAG interface (*below*)

Figure 15 - Other Examples of M3 JTAG Emulation Hardware

4.4 BNC Coax Sockets

The **4** x BNC bayonet sockets situated in the middle central area of the NetFusion-EXP baseboard allow for a screened coax cable transmission of the audio line IN and audio line OUT function. Whilst these are already provided as 2 x *3.5mm phono jacks* elsewhere on the board, the BNC feature allows for either oscilloscope analysis for clean noise free observation or for critical clean analogue waveforms to be shielded from surrounding industrial environmental interference.

All of the BNC sockets are labeled on the PCB so installation engineers or development programmer can access and identify which channel and direction of the audio with each connector. For audio line IN, there are stereo left and right channels, and the same with the audio line OUT.

4.5 Power Supply

NetFusion has two screw terminal DC power inlets and a single molded power receptacle. They are electrically all in parallel, but all have protection for differing voltages and overvoltage. Moreover, the polarity cannot be reversed so as to damage the baseboard either.

It is generally the case that the power receptacle is commonly used on the desk and during development. The screw terminal header inlets are for the industrial deployment arena and maybe the NetFusion-EXP is sharing power with other equipment or 3rd party bespoke DC power supplies are used. There are two screw terminal connectors in parallel to assist in the frequent and common installation practice of daisy chaining power together for multiple devices. Moreover, it can also be used if two power supplies are required to provide the current needed

for NetFusion-EXP on a demanding high-bandwidth switching application.

All accidental reverse polarity connection attempts are diode protected, along with over-voltage transzorb clamp protection.

All three inputs can accept a wide voltage range between 9V – 36V DC. The higher the voltage the lower the current sourced for the DC-DC module on the underside of the NetFusion-EXP. **Please Note:** *a standard **18V** soap-on-a-rope desk AC power supply is preferred for the receptacle inlet as this is the most optimum voltage for noise reduction for these types of supplies.*

Figure 16 - A Standard typical 18V DC Desk Mains Supply

4.6 Reset Button

The main reset button on the NetFusion-EXP board controls the reset sequence across the hardware. By pressing and releasing, an IP core in the FPGA fabric initiates a wave of reset flow that accurately brings up the Cortex-M3, FPGA fabric, SOM hardware and the NetFusion baseboard devices.

There is no ability to distinguish different desirable resets. It's the whole environment as one. You cannot just reset the Cortex-M3 only for instance and also want to independently reset other parts of the hardware on different occasions. The reason for this is there is only one reset button and trace line routed into the SOM then into the SmartFusion2 FPGA.

A solution to having NetFusion-EXP provide multiple reset signals is to use some of the available hardware on the baseboard. For instance, using the LCD buttons and re-wiring the connection inside the FPGA fabric using *Libero IDE*. Moreover, allowing the software to control output GPIO that is then re-wired in the fabric.

4.7 Generic Option Switches and Configuration Dials

Use the option switches and the rotary switches to control and bootstrap your firmware application code. All of these signals are routed to the GPIO core inside the SmartFusion2 FPGA fabric which is memory addressable from the ARM Cortex-M3 processor.

This enables software application code to use one of the UIO device drivers to ascertain the state and configuration of the option switches and the rotary switches. For instance, you can set parts of a MAC address or change fundamental MII modes.

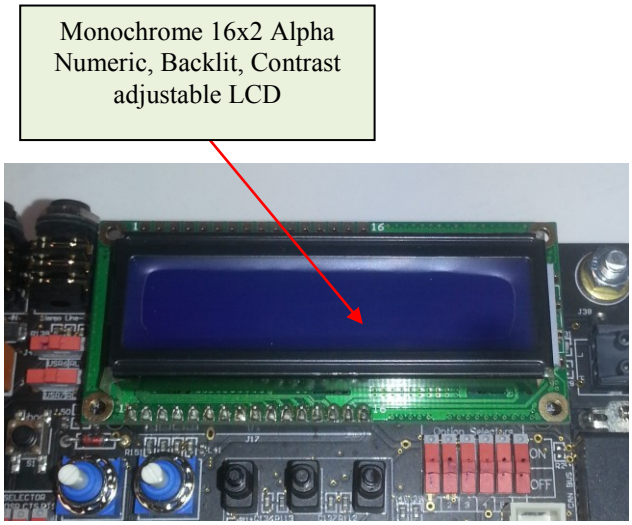
The most common usage is for user application program specific initialization settings.

4.8 User Buttons & LCD

The LCD is not fitted as standard to a NetFusion-EXP baseboard unless you have purchased the development kit bundle.

It is a 16 x 2 digit alpha-numeric screen with just black monochrome as a color. A rotary dial controls the contrast and also another dial controls the backlight power. The contents of the screen are controlled by application software via the GPIO core in the SmartFusion2 FPGA fabric.

The contents of the screen survive a hardware reset which is why it must be cleared and re-initialized upon software boot-up if the screen is to re-establish an initialized message announcement. The customer/user can use the screen (*if required*) to show whatever content they desire and is only designed in for user purposes. The LCD buttons are just input signals available to the application code as well.



4.9 Stereo Audio Phono-Jacks

The two stereo *3.5mm* phono jacks provide for audio line IN and audio line OUT. The usage can obviously go beyond audio as the sampling rate of the DAC; ADC devices can extend as high as 41.5kHz. The SPI DACs sample at MHz but in practice the software pipelines for the data can only cope with kHz.

Figure 17 - 3.5mm Phono Plug (3-pin)

Analogue waveforms incoming through both left and right channel of the audio line IN are sampled by an ADC. This is polled and read over SPI by application software. The same happens in reverse for the audio line OUT left and right channels.

Line OUT impedance: **100ohms**.
Line IN impedance: **8K2**

Note: they *appear to the uCLinux environment as pipes of data streams from the UIO audio drivers.*

4.10 Temperature Sensor

This device does not appear as a connector on the NetFusion-EXP baseboard. Specifically, it is an on-board component that checks and monitors the state of the PCB.

The temperature sensor IC is located on the surface of the PCB where the temperature is likely to be the hottest. This can be used by the application code and the user/customer to check for overheating and send warnings to a control station for instance.

The temperature IC sends a square wave signal with an even duty cycle that has a cyclic Hz rate equal to the temperature in Kelvin degrees. So *273Hz* is freezing point of water and *373Hz* is *100* Celsius and this relationship continues uniformly theoretically with-in the operating limits of the industrial temperature range.

The output trace from this device routes into the SmartFusion2 FPGA fabric where a counter allows user software to address how many cycles per second is occurring. From this, the UIO temperature device driver can interface with user application programs.

4.11 Voltage Monitor

This device does not appear as a connector on the NetFusion-EXP baseboard. In fact, it is an on-board component that checks and monitors the state of the PCB.

An I2C device allows most of the power rails to feed in as inputs and an analogue reading is taken. A software UIO device driver reads the values over I2C and applies simple mathematics to allow the voltages to be revealed to the customer/user application programs.

This can be used by the user/customer to check for malfunction or sever over-voltage conditions and send warnings to a control station for instance.

4.12 Real-Time Clock

Although the SmartFusion2 FPGA has an in-built clock timer module in the MSS, surviving power down and maintaining good accuracy are not guaranteed. Therefore, NetFusion-EXP has incorporated a battery backed-up **R**eal-**T**ime-**C**lock (*RTC*) function directly on the baseboard.

The battery is capable of maintaining power to the RTC device for 3 years approximately without the board being powered. The Battery charges when power is present.

The RTC device is accessed via the FPGA fabric using an I2C bus. A UIO Linux device driver accesses the device and presents a time and date to user application programs or the Linux BASH script environment.

This function can maintain a ticking time during power off and runs of a *32 kHz* crystal maintaining a good level of accuracy. If your *Libero IDE* NetFusion-EXP project has the on-board MSS RTC enabled, then that will be able to run in parallel and independently of this RTC hardware on the baseboard.

4.13 RS232 Serial Interface

The first UART interface of the uCLinux system in the Cortex-M3 processor is used for the USB-UART (*as described early in this section*). This presents itself as */dev/ttyS0* in the uCLinux file system of devices available to use. However, for */dev/ttsS1* (*the second serial port*) NetFusion-EXP exposes this as an industrial DB9 RS232 interface in parallel with an adjacent screw terminal header with the exact same RS232 level pins. The NetFusion-EXP uses an RS232 level-shifter to convert from the LVTTL of the SOM, and also provides 1000V isolation from induced voltages on the RS232 lines. This is very useful in the industrially deployed arena where typically NetFusion-EXP may be installed.

The normal RX and TX signals are routed directly to the DB9 and the screw header terminal for statutory use but the extra DTR, DSR, CTS, RTS, RI signals go through selector switches (*described early in this document*), before potentially routing to the DB9 headers.

The purpose of the extra LVTTL signals being switched is to give the user a choice of whether these signals are needed. They can be routed instead to the GPIO auxiliary interface to be used for other customer intended functionality. Most commonly RS232 only uses RX and TX so by switching the extra signals to the GPIO, it allows for the pins from the SOM to be re-used for other features. These extra DSR, DTR, RTS, CTS signals are not controlled in the MSS ARM UART but from the general GPIO driver in the NetFusion Linux system.

The signals excluding the RX and TX route back through the SmartFusion2 FPGA fabric through a UART core which is addressable as memory from the UART device driver. However, the RX and TX are connected direct to the MSS UART.

4.14 RS485 Serial Interface

The USB and RS232 ports are not the only UART functions available to the user. Another driver has been installed to allow for an RS484 interface for much longer distances of legacy communication. As the baseboard has been designed to allow for a lot of legacy interfacing including bringing modern high speed Ethernet communications alongside old infrastructure that needs supporting, it is statutory that RS485 is at the heart of the system.

Just a single screw terminal header brings the **R+, R-, T+, T**- 5V differential pair signal levels onto the NetFusion-EXP baseboard in which they route through *2000V* isolation protection in a level shifter. LVTTL connects the level shifter IC with the SOM whereby they route through the SmartFusion2 FPGA fabric through a UART core which is addressable as memory from the UART device driver.

Unlike the RS232 interface, there is only RX and TX functionality which are all handled by a single device driver which streams the received and transmitted data from the uCLinux environment.

MAX Distance End-to-End	BAUD
125m	230400
300m	115200
900m	57600
1800m	38400
3000m	9600
5500m	1200

Table 2: RS485 Distance capabilities with respect to BAUD rate

4.15 Voltage Free Relays

The NetFusion-EXP hosts two mains voltage relays that can be used for user general purpose activity. Each has a separate screw terminal connector for ease of installation and deployment and the voltage rating of such a switched secondary voltage is mains AC (*110/230V*). Although the current throughput is limited to much less than *1000mA*, this mains switched voltage can be used to control further large industrial control beyond and away from NetFusion-EXP. The isolation voltage for safety purposes is rated at *3500V* if transient or surge spikes affect the relay wiring. In fact, the baseboard houses an inner solid-state relay inside of the mechanical set to further protect the SOM device controlling the whole PCB.

Control of the relays is obtained with an uCLinux device driver for the GPIO. The **ON** or **OFF** state is driven through a core in the FPGA fabric which then routes to a set of selector switches on the PCB located in close proximity to the relays on the PCB. These selectors allow for the relay signals to be used in the GPIO interface instead if the relays are not required.

If the relays are selected then the signals connect to the inner solid-state relays then to the main mechanical set. **Note:** *normally open* ***(NO)*** *and normally closed* ***(NC)*** *controlled by other selector switches*.

4.16 Analogue Contact Inputs (ACI)

The most un-talked about feature of NetFusion-EXP is actually one of its best functions it can provide the user. There are **8** screw terminal connectors that individually each allow for an analogue input. The connectors are all individual as this makes wiring and interfacing in an installed and deployed environment especially in tight or restrictive spacing. Each input is biased with a *5V* pull-up on the positive pin and GND on the other. This voltage applied input allows for relays of other equipment to close or keep open the input as a form of legacy communication – and furthermore, the analogue orientation allows for parallel and series tamper resistors to be used to detect sabotage attempts.

Other uses for these inputs could be to directly measure and monitor scaled proportional voltages of 3rd party connected equipment. These inputs are compatible with other voltage applied equipment that assert their own 5V reference voltage as NetFusion-EXP only uses a weak pull-up resistor to its own +5V.

Note: *It is feasible to use these as high frequency sampling of the voltages. However new device drivers will have to be implemented in the uCLinux system.*

Each of the **8 A**nalogue **C**ontact **I**nputs (*ACI*) is read by an ADC device on the PCB which is accessed from the SmartFusion2 fabric via an I2C bus. The core in the fabric is GPIO which is memory addressable from the Cortex-M3 processor. This allows our uCLinux device driver to read the voltage readings and present them to user application code.

It is for the user to decide how to use these input when they tailor NetFusion-EXP as their own product and the possibilities of usage are widespread such as linking to the SQLite database in our uCLinux system and then displaying on a webpage for instance.

Important Note: *Due to sensitive analogue input levels, clamping and in-line protection to unwanted harsh electric voltage spikes, transients etc are not provided for this function. It is for the user to install their specific surge and noise suppression if it is deemed necessary.*

Critical: *Do not expose more than 5.5V to these inputs absolute maximum rating.*

4.17 Earth and Lightning Protection

Power supplies to industrially installed components should be robust and fully protected against short-circuit and overvoltage conditions, including lightning strikes. NetFusion-EXP's legacy interfaces all have heavy-duty lightning protection devices which protect the main system board from lightning strikes, but to ensure maximum protection, the earth terminal spade on the baseboard must be connected to a good-quality earth located close by using a deep spike or lattice.

4.18 LED Indicators

There are two RED power LEDs that indicate 2.5V rail and 5V rail levels. From this, diagnosis of a faulty baseboard is much easier if it is visual as to whether the power regulators and the DC-DC module are to blame. Dim RED power LEDs can also indicate a near-short or faulty component with the NetFusion-EXP PCB.

A GREEN user LED is controlled fully by the application programs via the GPIO UIO device driver. This is for general purpose usage by the user/customer.

A set of RED RX and TX LEDs are driven from the *FL232 USB-UART* device on NetFusion-EXP directly. These flicker and blink when data is communicating on the USB UART and this can be used to diagnose if the user is having problems with talking to the NetFusion Linux environment from their PC.

Note: *The USB-UART RED LEDs are driven direct from the USB device on the PCB and not the software application code.*

4.19 NetFusion-EXP Specific Interfaces

The main difference of the EXP variant against the CAT5 and SFP is not only that the Marvell 1Gbit/s Ethernet feature has been removed, but additionally that a dual Hirose expansion pluggable daughter card slot and secondly, a CAN transceiver with connectors has been added.

4.20 CAN Dual Transceiver

The NetFusion-EXP hosts the capability to allow the CAN Controller in the Smartfusion2 MSS to connect out to the real-world via a baseboard CAN transceiver. The device is a dual bus capable IC which enables independent buses to be controlled and managed by the CAN controller in the FPGA. Only one bus individually can be accessed and driven at any given time – this is controlled by GPIO enable lines visible to the ARM and directly strobes the CAN transceiver.

The CAN buses are protected by transzorb clamps to allow for the NetFusion-EXP to be exposed to the more harsh industrial environments especially ESD. The interfaces are two removable screw terminal headers for wired access **(J35, J56)** and two DB9 connectors **(J37, J57)** for a direct legacy use. These both connect to an *AMIS-42770 SM IC* device **(U13)** Dual high speed CAN transceiver. This transceiver then connects to the fabric in the FPGA on the SOM via tracks to Hirose connectors **(J12, J13**).

4.21 Dual Hirose Expansion Daughter Card Slot

Two Hirose DF40 connectors **(J27, J31)** along with for fixing support holes and a cleared non-populated area of the NetFusion-EXP baseboard allow for daughter card expansion PCBs to be affixed. There is one standard GPIO daughter card that is available from Nine Ways R&D and provides just basic I/O access with Zener diode protection. However, a customer/user can develop bespoke daughter cards.

The Altium PCB starter design is available from the Nine Ways website in the NetFusion product section. This allows the user to start with the correct board shape and Hirose connector positions and reduces the time for development and lowers the risk of first time mistakes in the design process.

Earlier in this document, tables show the pin-out usages of the Hirose expansion slots connectors, but the Altium starter design and the availability of the standard I/O Nine Ways daughter card PCB are available from the NetFusion product page at ninewaysco.uk

The connectors are *80-pin, 0.4mm pitch receptacles*, providing *4mm stacking height* implying that the daughter board may fit low on the base-board PCB or on the underside of the NetFusion-EXP baseboard if necessary.

All power supply levels, I2C buses, SPI buses, JTAG, spare I/O from the SOM headers, Ethernet channels, old expansion I/O, GND interleave, switches, buttons, jumpers and all other parallel useful signals are routed through the expansion Hirose connectors for daughterboard functionality and general usage.

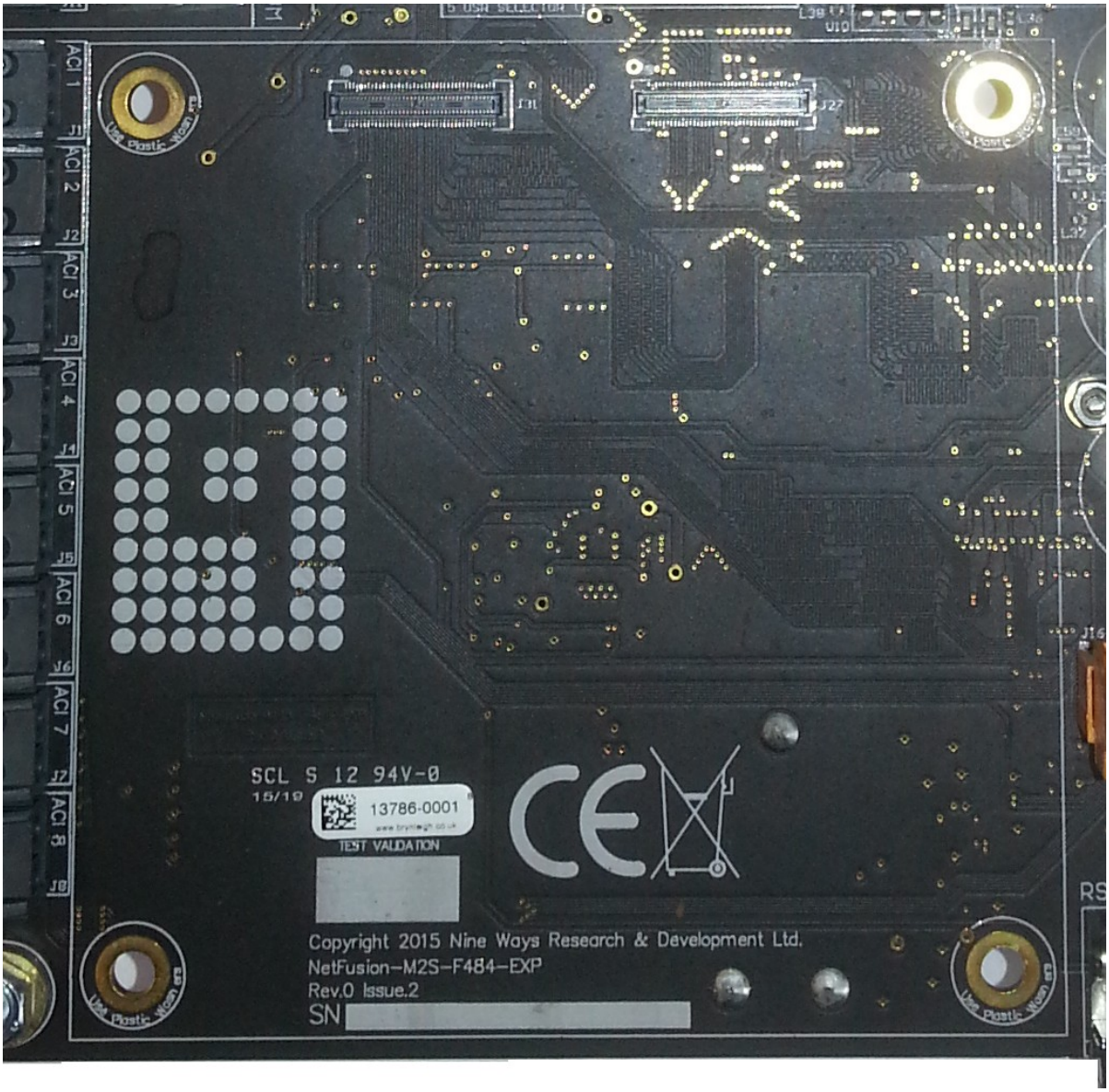
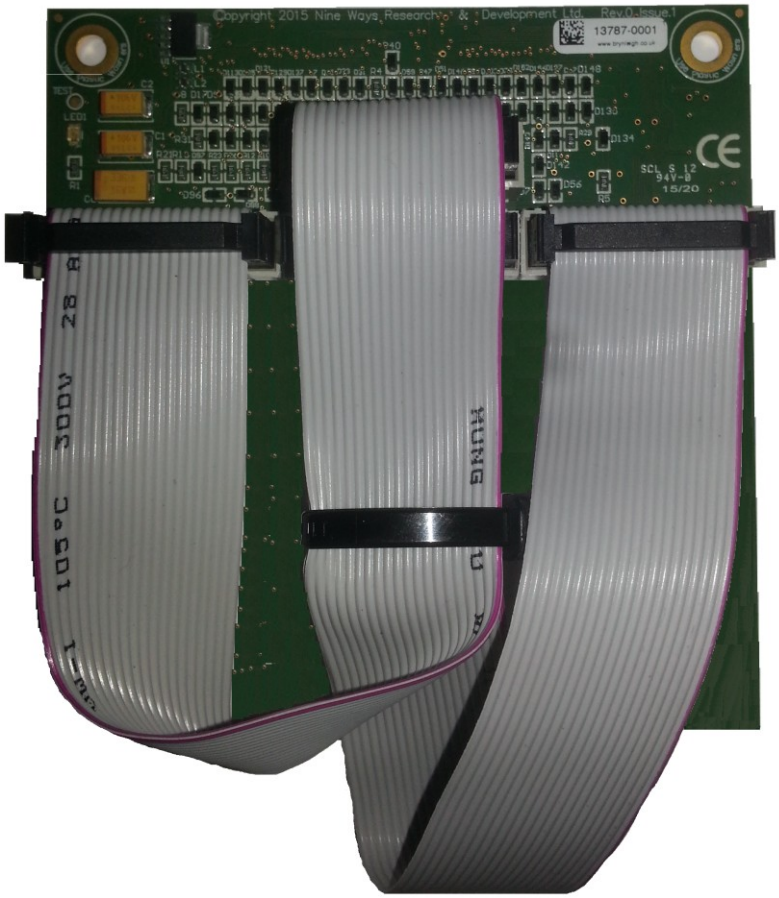


Figure 18 – Expansion Slot Area

5 Expansion I/O Daughter-Card



The EXP-daughterIO PCB product implements the following functionalities:

- *Four 20-way IDC polarized connectors for customer real-world interfacing*
- *Two 80-pin SOM connectors.*
- *Full ISO EN-61000:2011 EMI/EMC compatibility for electrical noise emission/immunity for both radiated and conducted categories.*
- *Industrial temperature capability of -40C to +85C degree Celsius for the main PCB.*
- *Full protection of the 2.5V I/O ball assignments of the SmartFusion2 050 F484 FPGA. Two IDC connectors are direction limited and two are input/output. The 2.5V signals are split into input IDC and output IDC as the conversion from 2.5 to/from 3.3V TTL is simpler.*
- *All signals are protected with 3.3V Zener diodes to stop over voltage to the FPGA. This protects accidental voltage damage and/or incorrect wiring from harming the silicon device. All voltage +-3.3V are clipped both positive and negative amplitudes and the current is dissipated via the EARTH conductive holes to the Netfusion-EXP baseboard.*
- *Red power LED to indicate the separately generated 3.3V is good.*

5.1 Daughter Card Design

The EXP daughterIO card is the first generic I/O expander for the Netfusion-EXP variant product. On its own, the NetFusion-EXP cannot be easily interfaced to via the expansion Hirose connectors from the baseboard. This is because the connectors have to be of high quality, low attenuation, high frequency and low noise as any wide range of usages can be created as an expansion card.

This EXP-daughterIO allows a simple 72 GPIO line interface from the SmartFusion2 FPGA on the SOM to be interfaced to the outside world via IDC headers. The headers will provide useful 3.3V and GND for pull-ups and GND referencing etc.

As the SmartFusion2 has different voltages for some banks of signals, some of the I/O is 3.3V and the rest 2.5V. This mixes *LVTTL* (3.3) with *LVCMOS25* (2.5) and over-volt aging an input can damage the FPGA over time. Moreover, the lines used as outputs at 2.5V may not drive a device needing 3.3V close threshold and certainly would not provide all of the necessary current. In some cases, *LVTTL* allows for 2.5V to drive 3.3V but this is not guaranteed and over time, signal distortion can occur with drift.

In order to get round this, one of the IDC connectors will have outputs only (*converted to 3.3V via reverse diodes and a pull-up*), and a second IDC as input only (*series drop resistors*). The remaining I/O IDC can be assigned as inputs or outputs.

For the two IDC that are designated as input ONLY or output ONLY, attempting to assign them oppositely in the Libero IDE will result in no physical damage but will produce unpredictable results. The output IC will not work as inputs at all due to the series diode. For the input IDC, trying to use them as outputs in Libero IDE may work if the target IC device can cope with around 1.8V threshold or less.

In general, keep inputs and output IDC signals dedicated and the rest can change or be randomly connected to the two I/O IDC connectors.

5.2 GPIO Hirose connectors

The EXP-daughterIO PCB houses two Hirose DF40 series connectors **(J1, J2)** for mating with the NetFusion-EXP baseboard PCB. The connectors are *80-pin, 0.4mm pitch plugs*, providing *1mm stacking height*.

Only the 5V and GN pins are used for power on the daughter card. All other power pins are not used as this board re-generated the 3.3V via a regulator so as not to load the existing 3.3V on the base-board. All of the 72 GPIO signals are connected via Zener clamping to protect the FPGA from over-voltage or incorrect connection.

5.3 Power Supply

The +5V from the baseboard is used to generate **(U1)** a new 3.3V on the daughter card. The incoming +5V are supplied from the DC-DC source on the NetFusion-EXP baseboard PCB and can deliver high current. The 3.3V on the daughter PCB then provides power to the IDC target board via the IDC ribbon cabling. This can be used for pull-ups for instance, or to power active circuit logic and devices.

5.4 Power Test

Test point **(J3)** above the RED LED allows for inspection of the 3.3V power rail from the daughter card regulator **(U1).**

5.5 IDC Interfaces

Connectors **(J4, J5, J6, and J7)** provide the simple customer interfaces. These can provide up to *10MHz* signaling with little attenuation or noise. Power and GND are provided and all signals **(GPIO)** are bi-directionally capable. The exceptions are **(J4)** output ONLY and **(J7)** input ONLY.

5.6 Earth Protection

The holes in the PCB either side of the Hirose mating connectors are the stabilizing and structural holes for screw and nut mounting. They are also conductive with the *EARTH* for power dissipation of *ESD* and low current induction. Metal washers and stand-offs are recommended here to the NetFusion-EXP baseboard PCB.

5.7 Daughter-Card Expansion I/O Connectors

5.7.1 Connector Summary

Table 3: Summary of PCB Connectors

Connector	Description
J1, J2	SOM header interface connectors
J5, J6	Input / Output IDC customer 20-way connectors
J4	Output ONLY IDC customer 20-way connector
J7	Input ONLY IDC customer 20-way connector
J3	3.3V test probe point

5.7.2 Customer IDC (Outputs ONLY) Connector (J4)

Table 4: IDC Connector J4 (Customer Interface)

Pin	Signal	Type	NetFusion Usage
1	*3.3V*	POWER	Separately generated 3.3V
2	*IOEXP24*	Output	SmartFusion2 050-F484 ball *N8* bank *5*
3	*IOEXP35*	Output	SmartFusion2 050-F484 ball *Y9* bank *5*
4	*IOEXP37*	Output	SmartFusion2 050-F484 ball *W9* bank *5*
5	*IOEXP39*	Output	SmartFusion2 050-F484 ball *Y10* bank *5*
6	*IOEXP41*	Output	SmartFusion2 050-F484 ball *W10* bank *5*
7	*IOEXP43*	Output	SmartFusion2 050-F484 ball *W11* bank *5*
8	*IOEXP45*	Output	SmartFusion2 050-F484 ball *V11* bank *5*
9	*IOEXP47*	Output	SmartFusion2 050-F484 ball *T13* bank *5*
10	*IOEXP64*	Output	SmartFusion2 050-F484 ball *M7* bank *7*
11	*IOEXP66*	Output	SmartFusion2 050-F484 ball *L7* bank *7*
12	*IOEXP70*	Output	SmartFusion2 050-F484 ball *AB15* bank *5*
13	*IOEXP72*	Output	SmartFusion2 050-F484 ball *AA15* bank *5*
14	*IOEXP42*	Output	SmartFusion2 050-F484 ball *H7* bank *8*
15	*IOEXP44*	Output	SmartFusion2 050-F484 ball *J6* bank *8*
16	*IOEXP68*	Output	SmartFusion2 050-F484 ball *K20* bank *3*
17	*IOEXP69*	Output	SmartFusion2 050-F484 ball *K21* bank *3*
18	*IOEXP71*	Output	SmartFusion2 050-F484 ball *E22* bank *1*
19	*IOEXP65*	Output	SmartFusion2 050-F484 ball *F20* bank *1*
20	*GND*	POWER	0V Chassis Ground

5.7.3 Customer IDC I/O Connector (J5)

Table 5: IDC Connector J5 (Customer Interface)

Pin	Signal	Type	NetFusion Usage
1	*3.3V*	POWER	Separately generated 3.3V
2	*IOEXP1*	I/O	SmartFusion2 050-F484 ball *K1* bank *7*
3	*IOEXP2*	I/O	SmartFusion2 050-F484 ball *K2* bank *7*
4	*IOEXP3*	I/O	SmartFusion2 050-F484 ball *M1* bank *7*
5	*IOEXP4*	I/O	SmartFusion2 050-F484 ball *N1* bank *7*
6	*IOEXP5*	I/O	SmartFusion2 050-F484 ball *P1* bank *7*
7	*IOEXP25*	I/O	SmartFusion2 050-F484 ball *K18* bank *3*
8	*IOEXP26*	I/O	SmartFusion2 050-F484 ball *B1* bank *8*
9	*IOEXP27*	I/O	SmartFusion2 050-F484 ball *B2* bank *8*
10	*IOEXP28*	I/O	SmartFusion2 050-F484 ball *C3* bank *8*
11	*IOEXP29*	I/O	SmartFusion2 050-F484 ball *A2* bank *8*
12	*IOEXP30*	I/O	SmartFusion2 050-F484 ball *C4* bank *8*
13	*IOEXP31*	I/O	SmartFusion2 050-F484 ball *F5* bank *8*
14	*IOEXP32*	I/O	SmartFusion2 050-F484 ball *E4* bank *8*
15	*IOEXP33*	I/O	SmartFusion2 050-F484 ball *D4* bank *8*
16	*IOEXP34*	I/O	SmartFusion2 050-F484 ball *F5* bank *8*
17	*IOEXP36*	I/O	SmartFusion2 050-F484 ball *G6* bank *8*
18	*IOEXP38*	I/O	SmartFusion2 050-F484 ball *F6* bank *8*
19	*IOEXP40*	I/O	SmartFusion2 050-F484 ball *G7* bank *8*
20	*GND*	POWER	0V Chassis Ground

5.7.4 Customer IDC I/O Connector (J6)

Table 6: IDC Connector J6 (Customer Interface)

Pin	Signal	Type	NetFusion Usage
1	*3.3V*	POWER	Separately generated 3.3V
2	*IOEXP46*	I/O	SmartFusion2 050-F484 ball *K7* bank *8*
3	*IOEXP48*	I/O	SmartFusion2 050-F484 ball *K6* bank *8*
4	*IOEXP49*	I/O	SmartFusion2 050-F484 ball *H6* bank *8*
5	*IOEXP50*	I/O	SmartFusion2 050-F484 ball *K15* bank *3*
6	*IOEXP51*	I/O	SmartFusion2 050-F484 ball *L16* bank *3*
7	*IOEXP52*	I/O	SmartFusion2 050-F484 ball *P16* bank *3*
8	*IOEXP53*	I/O	SmartFusion2 050-F484 ball *R16* bank *3*
9	*IOEXP54*	I/O	SmartFusion2 050-F484 ball *H16* bank *3*
10	*IOEXP55*	I/O	SmartFusion2 050-F484 ball *N16* bank *3*
11	*IOEXP56*	I/O	SmartFusion2 050-F484 ball *P18* bank *3*
12	*IOEXP57*	I/O	SmartFusion2 050-F484 ball *K17* bank *3*
13	*IOEXP58*	I/O	SmartFusion2 050-F484 ball *F18* bank *1*
14	*IOEXP59*	I/O	SmartFusion2 050-F484 ball *F19* bank *1*
15	*IOEXP60*	I/O	SmartFusion2 050-F484 ball *L18* bank *1*
16	*IOEXP61*	I/O	SmartFusion2 050-F484 ball *L19* bank *3*
17	*IOEXP62*	I/O	SmartFusion2 050-F484 ball *G19* bank *1*
18	*IOEXP63*	I/O	SmartFusion2 050-F484 ball *H20* bank *1*
19	*IOEXP67*	I/O	SmartFusion2 050-F484 ball *F21* bank *1*
20	*GND*	POWER	0V Chassis Ground

5.7.5 Customer IDC (Inputs ONLY) Connector (J7)

Table 7: IDC Connector J7 (Customer Interface)

Pin	Signal	Type	NetFusion Usage
1	*3.3V*	POWER	Separately generated 3.3V
2	*IOEXP23*	Input	SmartFusion2 050-F484 ball *P6* bank *7*
3	*IOEXP22*	Input	SmartFusion2 050-F484 ball *M5* bank *7*
4	*IOEXP21*	Input	SmartFusion2 050-F484 ball *K5* bank *7*
5	*IOEXP20*	Input	SmartFusion2 050-F484 ball *N6* bank *7*
6	*IOEXP19*	Input	SmartFusion2 050-F484 ball *K4* bank *7*
7	*IOEXP18*	Input	SmartFusion2 050-F484 ball *N5* bank *7*
8	*IOEXP17*	Input	SmartFusion2 050-F484 ball *L5* bank *7*
9	*IOEXP16*	Input	SmartFusion2 050-F484 ball *L3* bank *7*
10	*IOEXP15*	Input	SmartFusion2 050-F484 ball *L4* bank *7*
11	*IOEXP14*	Input	SmartFusion2 050-F484 ball *L2* bank *7*
12	*IOEXP13*	Input	SmartFusion2 050-F484 ball *M3* bank *7*
13	*IOEXP12*	Input	SmartFusion2 050-F484 ball *P4* bank *7*
14	*IOEXP11*	Input	SmartFusion2 050-F484 ball *M4* bank *7*
15	*IOEXP10*	Input	SmartFusion2 050-F484 ball *N3* bank *7*
16	*IOEXP9*	Input	SmartFusion2 050-F484 ball *N4* bank *7*
17	*IOEXP8*	Input	SmartFusion2 050-F484 ball *P3* bank *7*
18	*IOEXP7*	Input	SmartFusion2 050-F484 ball *P2* bank *7*
19	*IOEXP6*	Input	SmartFusion2 050-F484 ball *R1* bank *7*
20	*GND*	POWER	0V Chassis Ground

6 Hardware Device Driver Support

Netfusion-EXP currently has **11** x user-space hardware drivers. They are compiled as applications all separate to each other and are structured and designed such that they interface to the NetFusion-EXP PCB hardware via resource protected memory accessible FPGA IP cores.

The driver establishes the <*UioX*> interface between the hardware and the user-space. During normal operation, i.e. when the user-space application accesses the hardware, no additional software layers are involved. Memory accesses performed by the software are directly interchanged with the hardware.

Name of Driver	Source File	Description
Analogue Contacts (ACI)	*aci_ad7998.c*	Read analogue voltages from connector input
RS485 UART	*coreUART.c*	Data in|out through RS485 bus interface
Real-Time Clock (RTC)	*ds1307rtc.c*	Separate PCB baseboard battery-backed clock
LCD Controller	*lcd_pcf5074.c*	LCD optional display controller of text and format
General I/O (GPIO)	*pcf5074.c*	Individual spare pins and the LEDs, Switches & Jumpers
Line-OUT Stereo Audio	*spi_audio_line_out.c*	Stream digital audio to analogue Line-OUT audio
Line-IN Stereo Audio	*spi_audio_line_in.c*	Line-IN audio analogue to digital output stream
PCB Temperature Sensor	*temp_sensor.c*	An on-board temperature sensor on the baseboard
Voltage PCB Supervisor	*vs_ad7998.c*	Monitor and check all voltages and rails on the PCB
Expansion GPIO Controller	*ioexp.c*	Assert outputs and read inputs from the IOEXP
CAN Controller/Transceiver	*coreCAN.c*	Data management and test of the CAN buses

Table 8 : Full NetFusion User Space Hardware Drivers

Note*: the source files are located on your PC installed uCLinux cross-compile environment in the netfusion/source directory alongside the DMA Controller device driver source and the shell scripts.*

For a fully extensive and detailed description of the entire user-space hardware drivers of NetFusion-EXP as a Development-Kit, please refer to the **NetFusion Software uCLinux User Guide PDF.**

7 SQLite Usage

The SQLite is a specifically ported uCLinux orientated MySQL query binary application. As discussed, the source code compiles to the binary using *make*. However, once the user is running and using in the target NetFusion uCLinux system, a whole level of support and documentation is needed for those users who are not as familiar with the service.

Please refer to the following documentation: SQLite Official Website

Also, type *sqlite3 –help* on the NetFusion prompt for full descriptive options and usage as a command helper.

8 NetFusion-EXP Development-Kit Web-Server

The NetFusion-EXP default uCLinux bundle has a working and ready-to-run *apache2* web-server. This is compiled and guised in the form of *httpd* from *busybox*. The */httpd/html* directory contains the web page echo-system where you can change and re-design to you own desires and architectural requirements.

As a start point in the bundled NetFusion uCLinux Development-Kit package, the web page system shows a demonstration of the PCB hardware and the SQLite capabilities. Also, the NetFusion user-space UIO drivers are all used to facilitate the web content which can aid and help the user see how the *cgi* mechanism by default, can allow hardware to interact with the web front end. Moreover, you can then grasp the value of the SQLite database system be able to use the hardware and the web-page to create a truly useful product which all takes advantage of the */flash* mounted SPI hardware using the **JFFS2** partition to allow for the persistent storage.

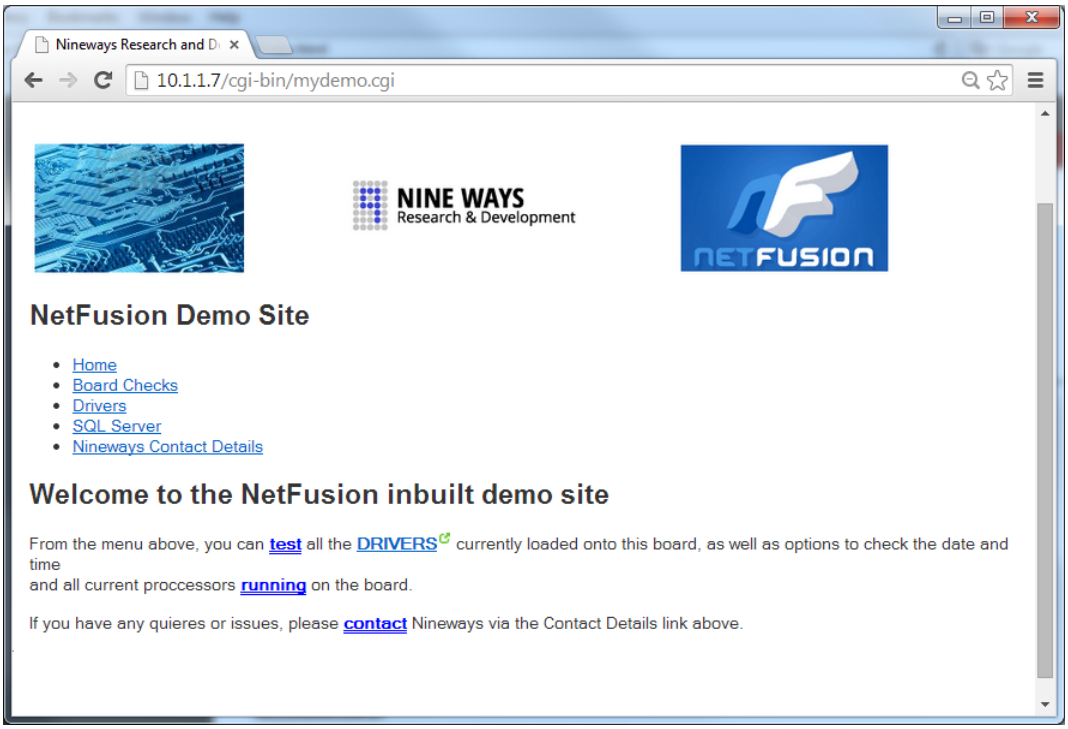

Nineways Research and D
10.1.1.7/cgi-bin/mydemo.cgi
NINE WAYS
Research & Development
NetFusion Demo Site
Home
Board Checks
Drivers
SQL Server
Nineways Contact Details
Welcome to the NetFusion inbuilt demo site
From the menu above, you can test all the DRIVERS currently loaded onto this board, as well as options to check the date and time
and all current proccessors running on the board.
If you have any quieres or issues, please contact Nineways via the Contact Details link above.

Figure 19 - Main opening screen for the NetFusion-EXP Demo Website

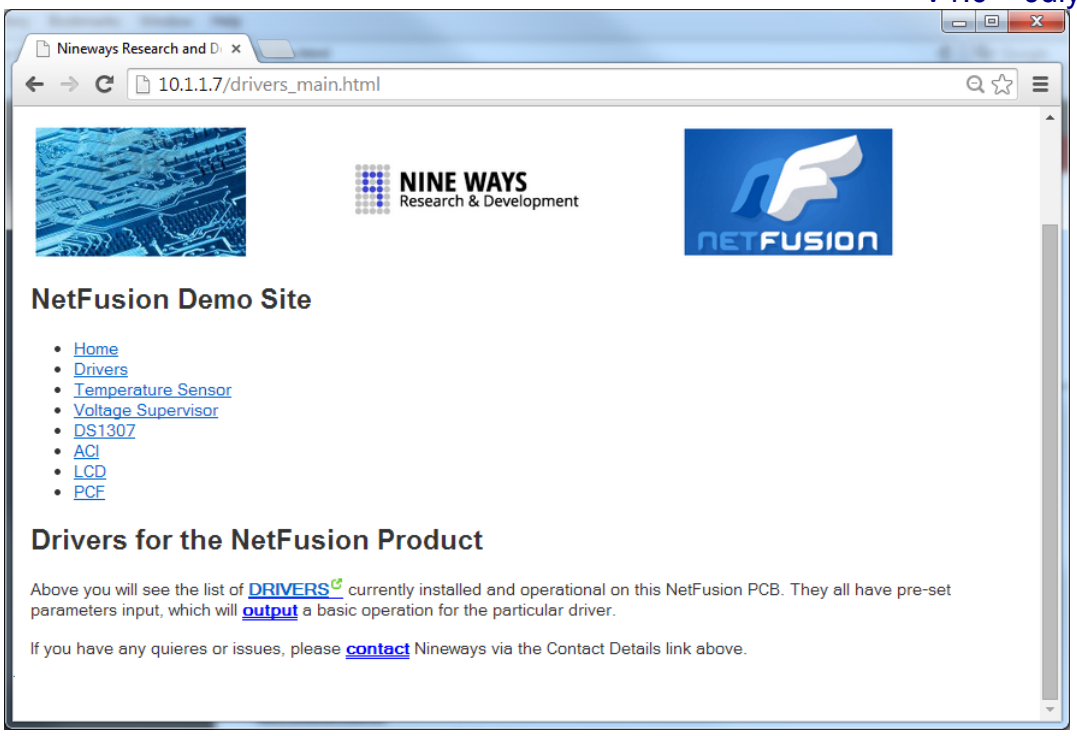

Nineways Research and D
10.1.1.7/drivers_main.html
NINE WAYS
Research & Development
NetFusion Demo Site
Home
Drivers
Temperature Sensor
Voltage Supervisor
DS1307
ACI
LCD
PCF
Drivers for the NetFusion Product
Above you will see the list of DRIVERS currently installed and operational on this NetFusion PCB. They all have pre-set parameters input, which will output a basic operation for the particular driver.
If you have any quieres or issues, please contact Nineways via the Contact Details link above.

Figure 20 – Drivers/Hardware Page for the NetFusion-EXP Demo Website

9 Demo Application

As part of all NetFusion-EXP Development-Kit products, the firmware code is shipped with an LCD demonstration application. This demonstrates most of the hardware and board - without having to compile or install any IDEs.

Once the user has installed all of the uCLinux cross-compile tools, they can remove the demonstration application from the LCD by commenting out the *demoapp* entry in *the local/rc* control boot script file.

It is ideal for an *out-of-the-box* play and familiarization with the features and functions of the NetFusion-EXP product. Eventually, the user will want to cut this out of the boot-up script file once the user wants to progress.

For information on how to operate and use the *LCD demo*, please refer to the **Demonstration Application Guide PDF**

CAT5e Ethernet Cable & Connectors

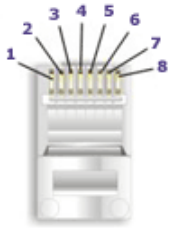
CAT5e cable connections should be made using RJ45 plugs. To comply with NetFusion-EXP's temperature range requirements, all cables and connectors must be specified to industrial standard, with an operating temperature range of *-40°C to +85°C.*

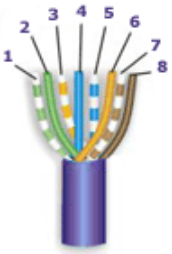
The maximum length of CAT5|6e connections specified in this document is *100m*. To operate reliably over distances close to this limit, high-quality cables and connectors should always be used. If the location of the cable is subject to large amount of electromagnetic noise, this may affect the maximum distance for cable runs.

9.1 RJ45 plug and CAT5|6e cable connections

Copper network connections within a NetFusion-EXP system must always be made using standard CAT5|6e straight-through cables. These can be wired using either *568A* or *568B* pin layout, but both ends of the cable should use the same pin layout.

568A Wiring





Pair	Wire	Pin number
Blue	Blue / White	4
	White / Blue	5
Green	Green / White	2
	White / Green	1
Orange	Orange / White	6
	White / Orange	3
Brown	Brown / White	8
	White / Brown	7

568B Wiring

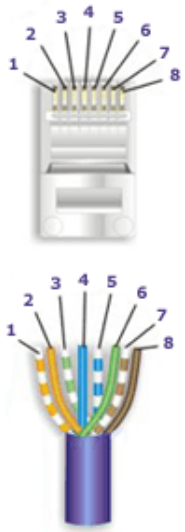
	Pair	Wire	Pin number
	Blue	Blue / White	4
		White / Blue	5
	Green	Green / White	6
		White / Green	3
	Orange	Orange / White	2
		White / Orange	1
	Brown	Brown / White	8
		White / Brown	7

Table 9 : Typical CAT5|6 cable wiring

10 SmartFusion2 FPGA NetFusion-EXP Fabric Development

10.1 Libero SoC/IDE Version 11.x Derivatives

Microsemi's Libero ® IDE software release for designing with Microsemi Rad-Tolerant FPGAs, Antifuse FPGAs and Legacy & Discontinued Flash FPGAs and managing the entire design flow from design entry, synthesis and simulation, through place-and-route, timing and power analysis. PCN 1108: Silicon Family Support in Libero IDE.

Libero IDE Software Features:

- Powerful project and design flow management
- Full suite of integrated design entry tools and methodologies:
  - SmartDesign graphical SoC design creation with automatic abstraction to HDL
  - IP Core Catalog and configuration
  - User-defined block creation flow for design re-use
- Synplify Pro ME synthesis fully optimizes Microsemi FPGA device performance and area utilization
- Synphony Model Compiler ME performs high-level synthesis optimizations within a Simulink® environment
- Modelsim ME VHDL or Verilog behavioral, post-synthesis and post-layout simulation capability
- Physical design implementation, floor planning, physical constraints, and layout
- Timing-driven and power-driven place-and-route
- SmartTime environment for timing constraint management and analysis
- SmartPower provides comprehensive power analysis for actual and "what if" power scenarios
- Interface to FlashPro programmers
- Post-route On Chip Debug Tools and Identify ME debugging software for Microsemi flash designs
- Silicon Explorer II debugging software for Microsemi antifuse designs

10.2 Installation

Libero software is downloadable for free from http://www.microsemi.com/products/fpga-soc/designresources/design-software/libero-soc#downloads. Some Libero features are optional during installation.

You can minimize the disk space required by only installing tools you use.

You must have a license to run Libero; the license type that you obtain determines what devices you can use and what IP is included. The following license types exist for Libero:

- Libero Platinum (*All devices and RTL IP Bundle*)
- Libero Gold (*Limited devices and Obfuscated IP Bundle*)

View the complete descriptions of the above Libero installations at http://www.microsemi.com/products/fpga-soc/design-resources/design-software/libero soc#licensing.

View the IP Bundle contents at http://www.microsemi.com/products/fpga-soc/design-resources/ip-cores.

Libero installation is covered in "Installing Libero Software" on page 16.

Note: You must have Admin rights on the installation machine to install Libero SoC.

10.3 Starting a Project and Basic Understanding

Before attempting to modify or implement any project in Libero, it is advised that you download and read the following PDF references:

System on Chip installation:

Libero SoC v11.X User's Guide

Libero SoC Quick Start Guide for Software v10.0

Integrated Development Environment installation:

Libero IDE and Software Installation and Licensing Guide

Libero IDE License Troubleshooting Guide

(**Note:** *Press CNTL and click to download the links*)

10.3.1 Libero 11.X IDE First Launch

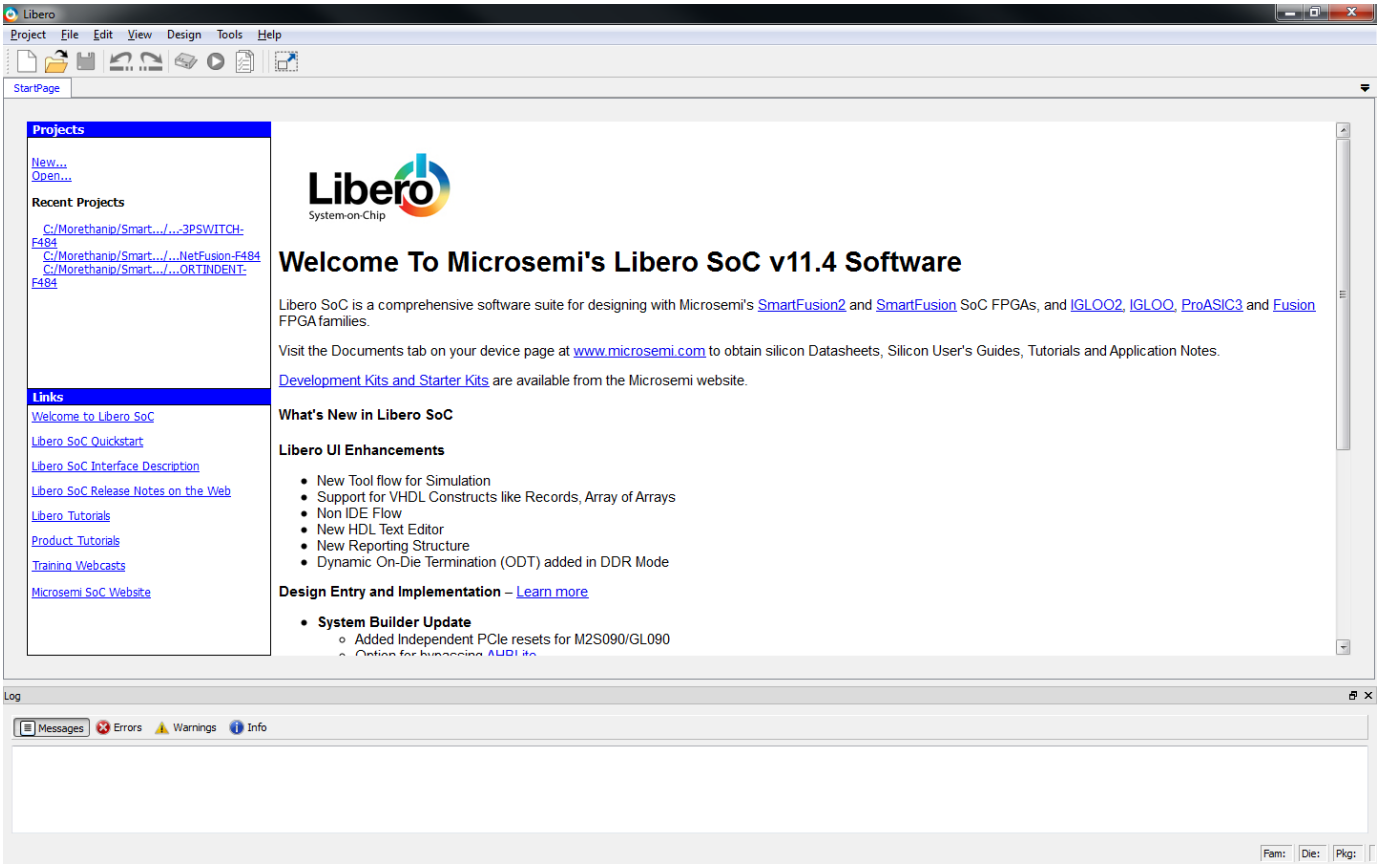

Libero
Project File Edit View Design Tools Help
StartPage
Projects
New...
Open...
Recent Projects
C:/Morethanip/Smart.../...-3PSWITCH-F484
C:/Morethanip/Smart.../...NetFusion-F484
C:/Morethanip/Smart.../...ORTINDENT-F484
Links
Welcome to Libero SoC
Libero SoC Quickstart
Libero SoC Interface Description
Libero SoC Release Notes on the Web
Libero Tutorials
Product Tutorials
Training Webcasts
Microsemi SoC Website
Libero
System-on-Chip
Welcome To Microsemi's Libero SoC v11.4 Software
Libero SoC is a comprehensive software suite for designing with Microsemi's SmartFusion2 and SmartFusion SoC FPGAs, and IGLOO2, IGLOO, ProASIC3 and Fusion FPGA families.
Visit the Documents tab on your device page at www.microsemi.com to obtain silicon Datasheets, Silicon User's Guides, Tutorials and Application Notes.
Development Kits and Starter Kits are available from the Microsemi website.
What's New in Libero SoC
Libero UI Enhancements
New Tool flow for Simulation
Support for VHDL Constructs like Records, Array of Arrays
Non IDE Flow
New HDL Text Editor
New Reporting Structure
Dynamic On-Die Termination (ODT) added in DDR Mode
Design Entry and Implementation – Learn more
System Builder Update
Added Independent PCIe resets for M2S090/GL090
Log
Messages
Errors
Warnings
Info
Fam:
Die:
Pkg:

Figure 21 - Libero 11.X IDE Boot-Up Screen

10.4 NetFusion-EXP

When purchasing any NetFusion-EXP variant, an important component of the overall product is the NetFusion-EXP Libero starter project that is provided by Nine Ways Research & Development Ltd. This is downloadable from Nine Ways R&D Ltd and when expanded into a target directory on your development PC, provides an immediate project for your needs. Instead of having to start and debug creating an ARM sub-system with all of the supporting IP cores required to have a functioning NetFusion-EXP PCB – just use the provided project.

From installation, you can program the IDE project once synthesized into your NetFusion-EXP product using the FlashPro4 USB device. This will provide the standard functionality in the Smartfusion2 FPGA fabric on the M2S-SOM-F484 **SOM** (**S**ystem-**O**n-**M**odule) to see a working project. In the software bundle to this product, the uClinux device drivers will drive and control the hardware on the NetFusion-EXP PCB through the IP cores in this project for the FPGA fabric.

Users can contact Nine Ways R&D for special functionality to be developed and deployed, but this serves as an addition to this starter Libero 11.X IDE project. Moreover, in conjunction with MorethanIP GmbH, customized and locked down projects can be tailored for customer requests but that also is separate to this project.

10.5 Product Development

At the point where the Libero IDE/SoC has been installed, the NetFusion-EXP starter project has been downloaded and exploded into a target directory on your PC, the project has been loaded, synthesized, programmed and shown to be running on the NetFusion-EXP PCB product – you are ready to begin your development.

As standard, the main fast Ethernet pathways into the SmartFusion2 FPGA fabric are wired in through the FPGA ball I/O, assigned in the I/O editor, brought down through the Top-Level and then into the SOM level of the design in the project. They then terminate at a dummy IP core for all un-assigned wires – this makes life a lot easier for the developer knowing that all the NetFusion-EXP traces coming into the M2S-SOM-F484 are wired into the SOM level of the fabric design. Changes are therefore quick and easy to then re-assign in that lower level sheet to new instantiated IP cores of the user's choice.

The category of wires left terminated and not used are the GMII Ethernet pathways. Users can download and use Vendor specific MAC/SWITCH cores or chose to privately purchase cores from reputable design-houses such as MorethanIP. All other used hardware on the NetFusion-EXP PCB is wired and connected in the SOM sheet layer to *GPIO, SPI, UART, USB, I2C* etc as standard in the starter project.

If you decide to write and author your own IP cores in Verilog or VHDL, you can drop and place the code into the **\[Project Dir]\hdl** directory.

Note: *Once you have started to customize and tailor the project to your own needs and functionality, obviously renaming the project is easy – just rename the project directory and inside that directory, just rename the **[.prj]** file. Close and re-open the Libero IDE.*

11 Downloading New IP Core(s) into the Libero IDE Installation

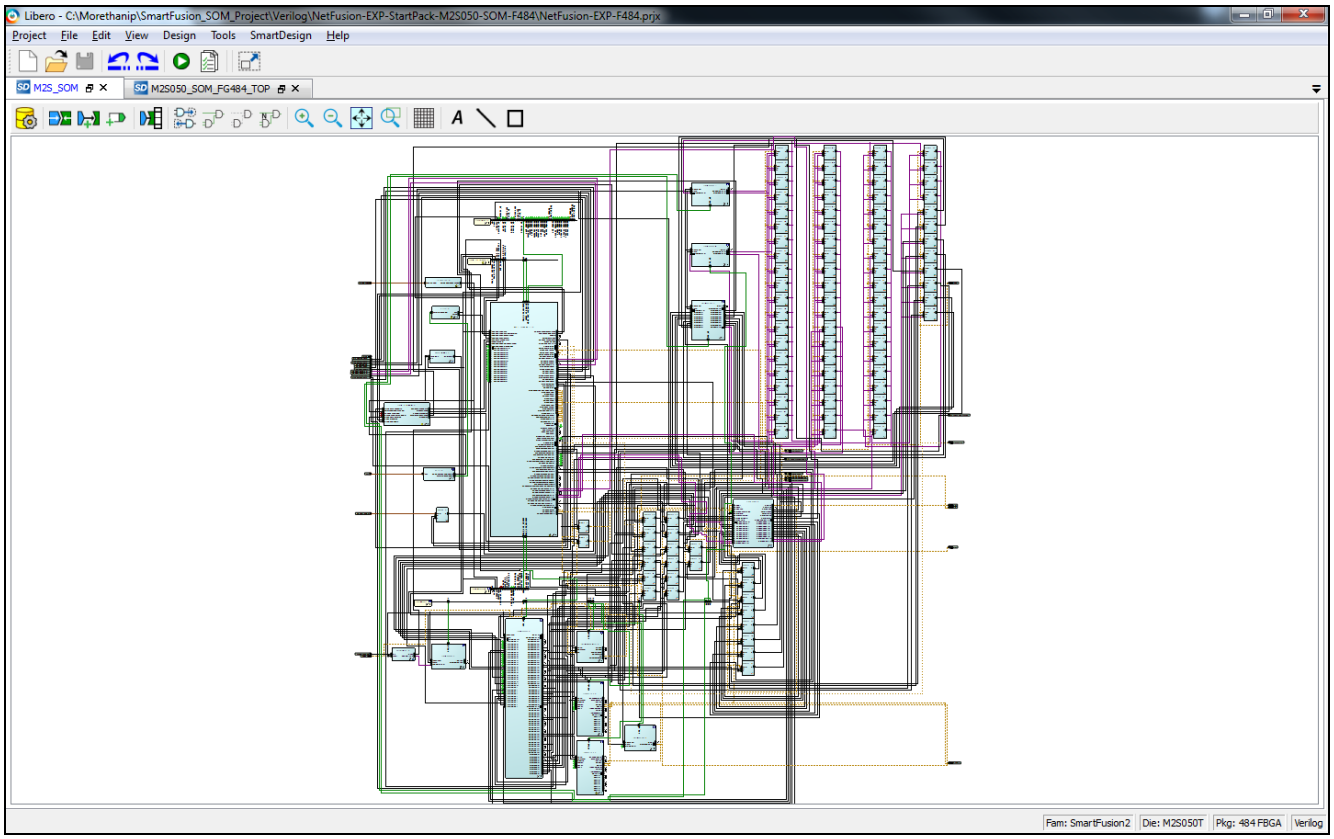


Figure 22 - Libero Updating IP Core in the Vault

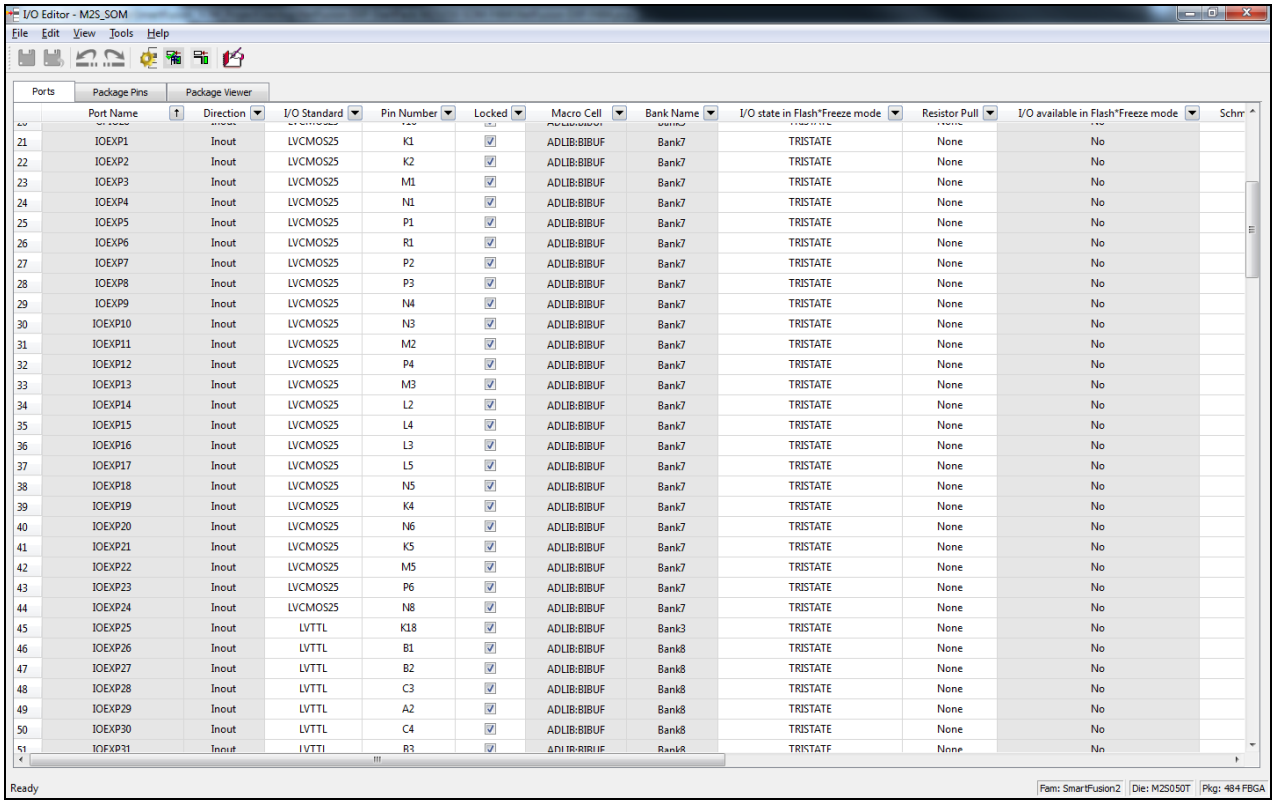
When you first load the NetFusion-EXP starter project into the IDE, the most likely occurrence to happen is a system message warning you to update “New IP Cores”. This is because the design includes cores that you possibly do not have on your vault on your local hard drive.

Click “*YES*” to proceed and let the download process complete. **Note:** *this may take several minutes and you MUST have a network connection and gateway to the internet.*

12 NetFusion-EXP Libero I/O Assignments

The NetFusion-EXP Libero starter project is delivered already with the I/O balls of the FPGA assigned ready for the user. These all correspond with the pin-out tracking assignments on the NetFusion-EXP PCB product. This means that you will not have to consider any of these assignments unless you are planning any major changes to the inherent default design.

This is very unlikely as you would need then to request changes to the PCB design. However, this section highlights the assignments in case you also want to make minor changes to the direction of the port, default output values and/or change internal SmartFusion2 FPGA pull-up values.



	Port Name	Direction	I/O Standard	Pin Number	Locked	Macro Cell	Bank Name	I/O state in Flash*Freeze mode	Resistor Pull	I/O available in Flash*Freeze mode
21	IOEXP1	Inout	LVCMOS25	K1	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
22	IOEXP2	Inout	LVCMOS25	K2	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
23	IOEXP3	Inout	LVCMOS25	M1	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
24	IOEXP4	Inout	LVCMOS25	N1	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
25	IOEXP5	Inout	LVCMOS25	P1	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
26	IOEXP6	Inout	LVCMOS25	R1	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
27	IOEXP7	Inout	LVCMOS25	P2	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
28	IOEXP8	Inout	LVCMOS25	P3	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
29	IOEXP9	Inout	LVCMOS25	N4	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
30	IOEXP10	Inout	LVCMOS25	N3	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
31	IOEXP11	Inout	LVCMOS25	M2	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
32	IOEXP12	Inout	LVCMOS25	P4	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
33	IOEXP13	Inout	LVCMOS25	M3	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
34	IOEXP14	Inout	LVCMOS25	L2	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
35	IOEXP15	Inout	LVCMOS25	L4	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
36	IOEXP16	Inout	LVCMOS25	L3	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
37	IOEXP17	Inout	LVCMOS25	L5	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
38	IOEXP18	Inout	LVCMOS25	N5	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
39	IOEXP19	Inout	LVCMOS25	K4	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
40	IOEXP20	Inout	LVCMOS25	N6	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
41	IOEXP21	Inout	LVCMOS25	K5	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
42	IOEXP22	Inout	LVCMOS25	M5	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
43	IOEXP23	Inout	LVCMOS25	P6	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
44	IOEXP24	Inout	LVCMOS25	N8	☑	ADLIB:BIBUF	Bank7	TRISTATE	None	No
45	IOEXP25	Inout	LVTTL	K18	☑	ADLIB:BIBUF	Bank3	TRISTATE	None	No
46	IOEXP26	Inout	LVTTL	B1	☑	ADLIB:BIBUF	Bank8	TRISTATE	None	No
47	IOEXP27	Inout	LVTTL	B2	☑	ADLIB:BIBUF	Bank8	TRISTATE	None	No
48	IOEXP28	Inout	LVTTL	C3	☑	ADLIB:BIBUF	Bank8	TRISTATE	None	No
49	IOEXP29	Inout	LVTTL	A2	☑	ADLIB:BIBUF	Bank8	TRISTATE	None	No
50	IOEXP30	Inout	LVTTL	C4	☑	ADLIB:BIBUF	Bank8	TRISTATE	None	No
51	IOEXP31	Inout	LVTTL	B3	☑	ADLIB:BIBUF	Bank8	TRISTATE	None	No

Figure 23 - Example of Top List I/O Ball Assignments

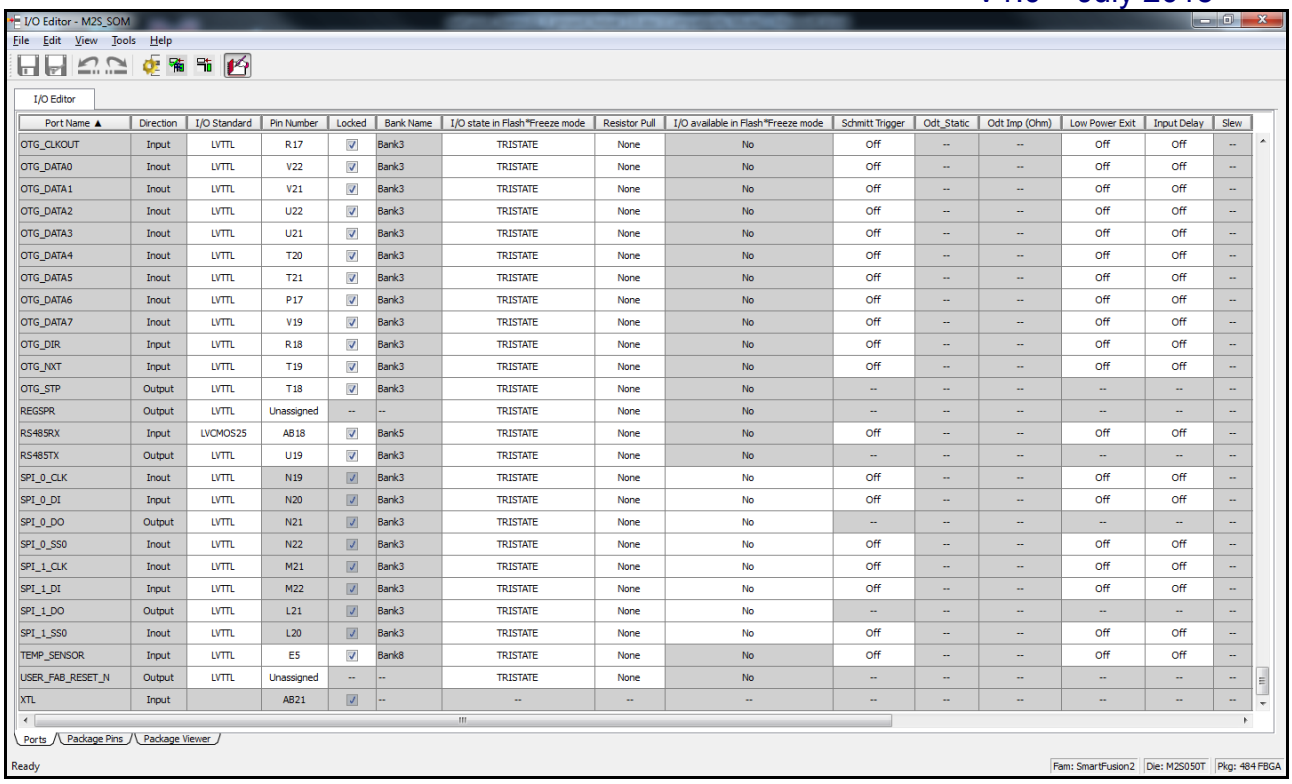

I/O Editor - M2S_SOM
File Edit View Tools Help
I/O Editor
Port Name ▲ | Direction | I/O Standard | Pin Number | Locked | Bank Name | I/O state in Flash*Freeze mode | Resistor Pull | I/O available in Flash*Freeze mode | Schmitt Trigger | Odt_Static | Odt Imp (Ohm) | Low Power Exit | Input Delay | Slew
OTG_CLKOUT | Input | LVTTL | R17 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_DATA0 | Inout | LVTTL | V22 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_DATA1 | Inout | LVTTL | V21 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_DATA2 | Inout | LVTTL | U22 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_DATA3 | Inout | LVTTL | U21 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_DATA4 | Inout | LVTTL | T20 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_DATA5 | Inout | LVTTL | T21 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_DATA6 | Inout | LVTTL | P17 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_DATA7 | Inout | LVTTL | V19 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_DIR | Input | LVTTL | R18 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_NXT | Input | LVTTL | T19 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
OTG_STP | Output | LVTTL | T18 | Bank3 | TRISTATE | None | No | -- | -- | -- | -- | -- | --
REGSPR | Output | LVTTL | Unassigned | -- | -- | TRISTATE | None | No | -- | -- | -- | -- | -- | --
RS485RX | Input | LVCMOS25 | AB18 | Bank5 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
RS485TX | Output | LVTTL | U19 | Bank3 | TRISTATE | None | No | -- | -- | -- | -- | -- | --
SPI_0_CLK | Inout | LVTTL | N19 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
SPI_0_DI | Input | LVTTL | N20 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
SPI_0_DO | Output | LVTTL | N21 | Bank3 | TRISTATE | None | No | -- | -- | -- | -- | -- | --
SPI_0_SS0 | Inout | LVTTL | N22 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
SPI_1_CLK | Inout | LVTTL | M21 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
SPI_1_DI | Input | LVTTL | M22 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
SPI_1_DO | Output | LVTTL | L21 | Bank3 | TRISTATE | None | No | -- | -- | -- | -- | -- | --
SPI_1_SS0 | Inout | LVTTL | L20 | Bank3 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
TEMP_SENSOR | Input | LVTTL | E5 | Bank8 | TRISTATE | None | No | Off | -- | -- | Off | Off | --
USER_FAB_RESET_N | Output | LVTTL | Unassigned | -- | -- | TRISTATE | None | No | -- | -- | -- | -- | -- | --
XTL | Input | | AB21 | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | --
Ports Package Pins Package Viewer
Ready
Fam: SmartFusion2 Die: M2S050T Pkg: 484 FBGA

Figure 24 - Example of Bottom List I/O Ball Assignments

The I/O assignment dialog is selected from the I/O Constraints anchor in Place and Route section (*below*)

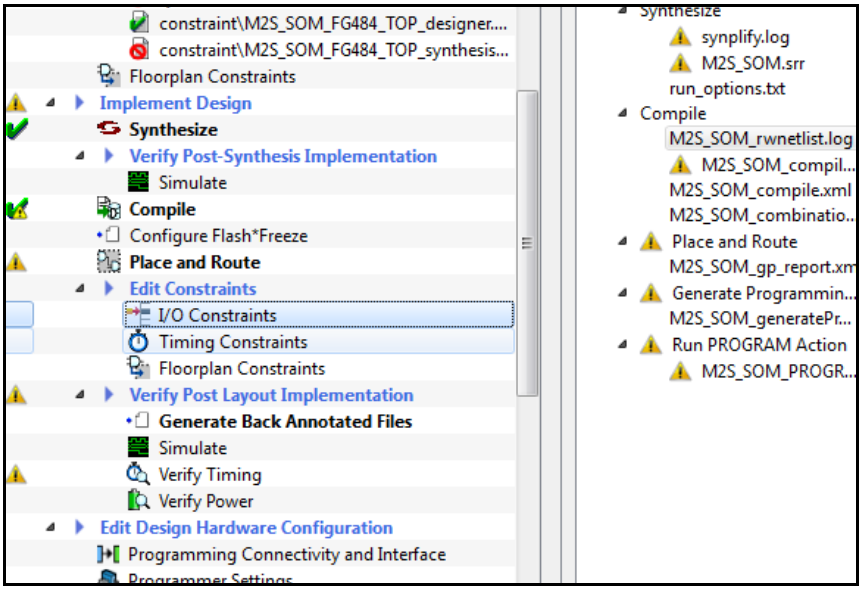


Figure 25 - Selecting the I/O Constraints

13 NetFusion-EXP Libero 11.X IDE/SoC Top-Level

Within the NetFusion-EXP starter Libero project, the M2S_SOM Top-Level sheet illustrates the lower modules to the design just before the I/O ports go out to the real-world PCB. The ports labeled in this sheet correspond to the I/O assignments (*shown in the previous section*). This sheet is effectively the linkage from the inner SOM design (*see next section*) up to the I/O balls of the SmartFusion2 FPGA.

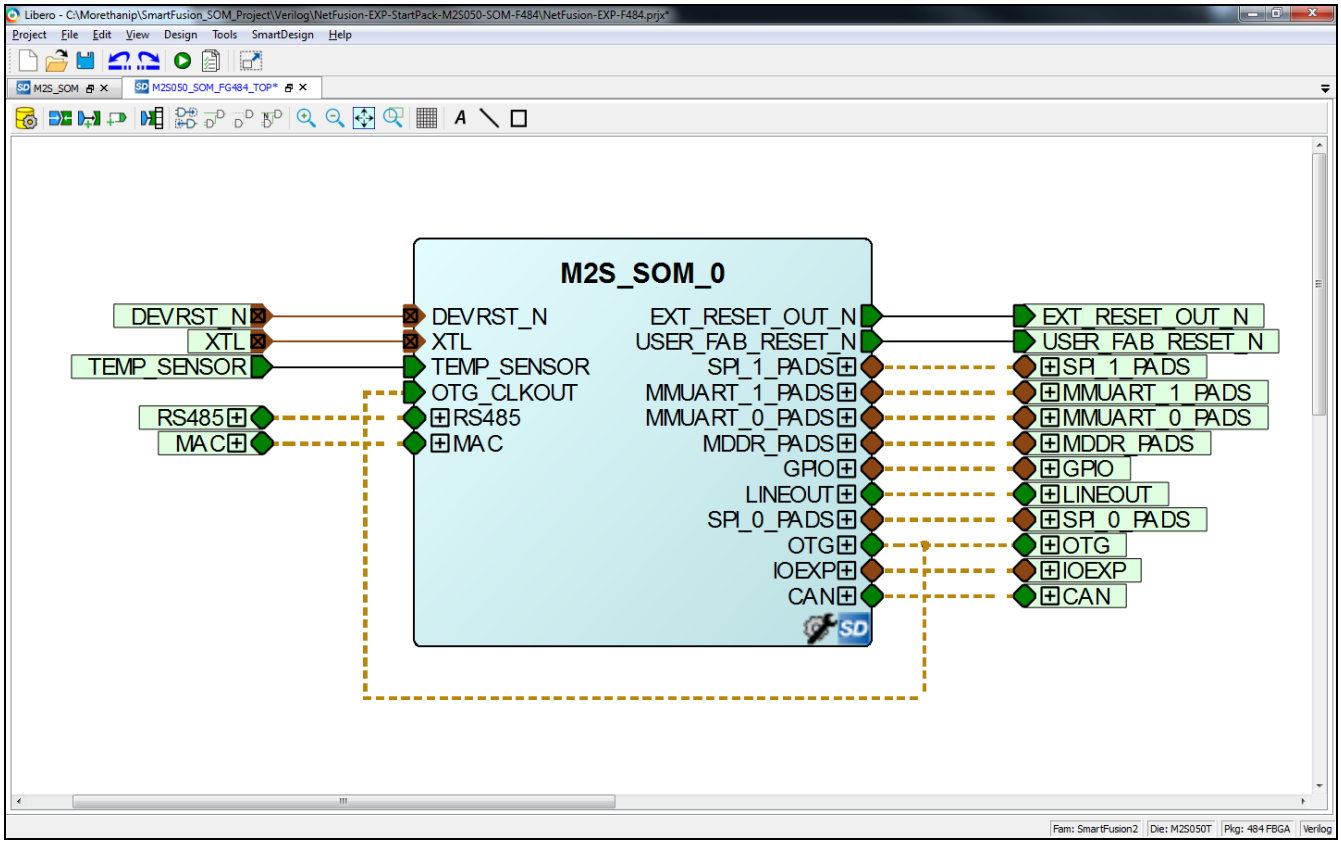


Figure 26 – NetFusion-EXP Top-Level Sheet

When lower level SOM sheets are built and prepared, they propagate information up automatically to this higher sheet. New ports suddenly appear and you must then compile this sheet before the main synthesis.

Note: *you can instantiate normal IP cores into this top sheet if you wish and it makes sense according to your design requirements. This is just the starter project so everything by default is kept in the lower SOM module. But this can change rapidly as your customization starts to take effect.*

The lower SOM sheet of the NetFusion-EXP starter Libero project contains all of the default designs and linkage.

This SOM sheet lower layer contains the ARM Cortex-M3 MSS block and then all of the necessary associated IP core blocks to allow the processor, the Emcraft SOM and the wider surrounding NetFusion-EXP baseboard hardware to operate and be accessed.

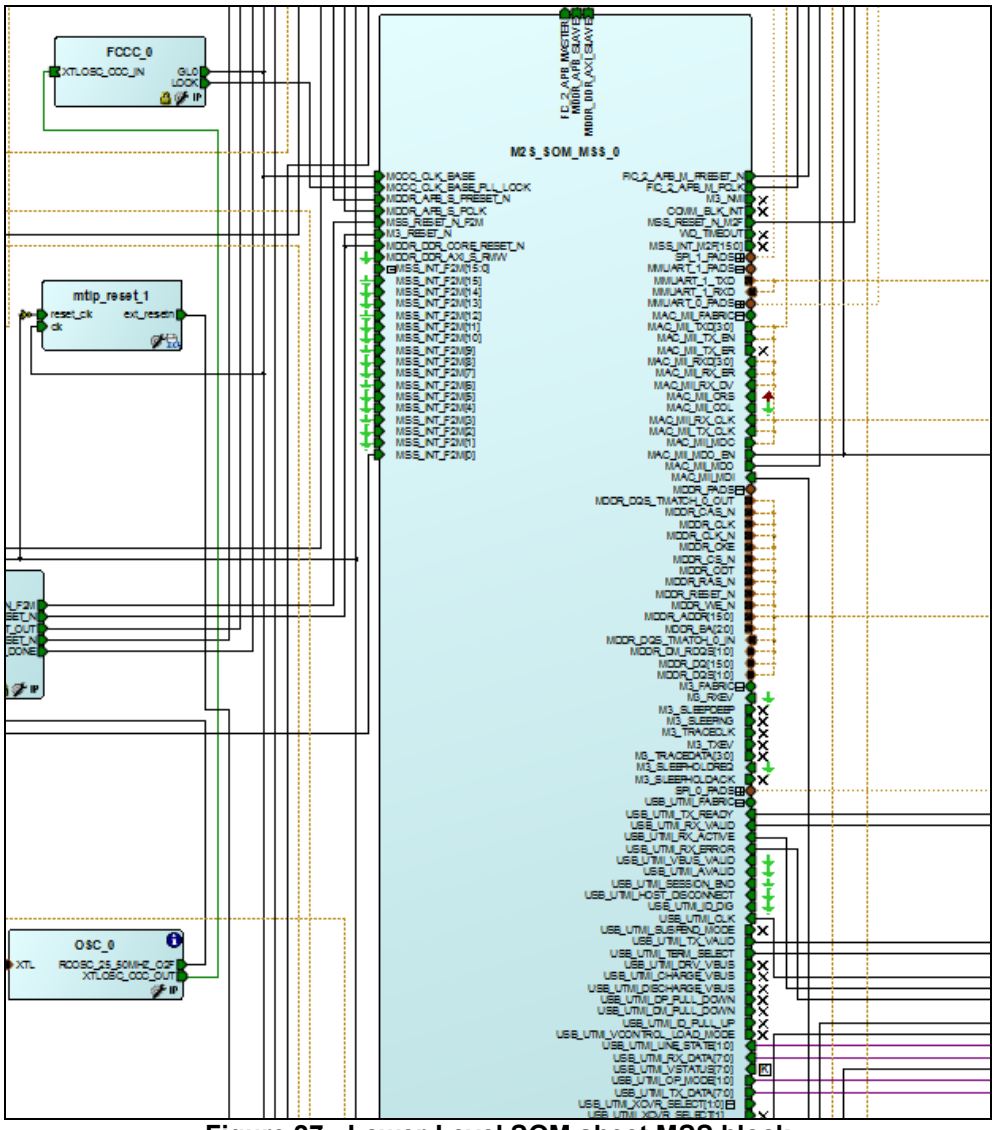


Figure 27 - Lower-Level SOM sheet MSS block

The next screenshot shows a rats-nest but the following sub-sections illustrate the different parts in more detail. **Note:** *Incidentally, the largest module center-left is the MSS ARM cortex processor (zoomed above).*

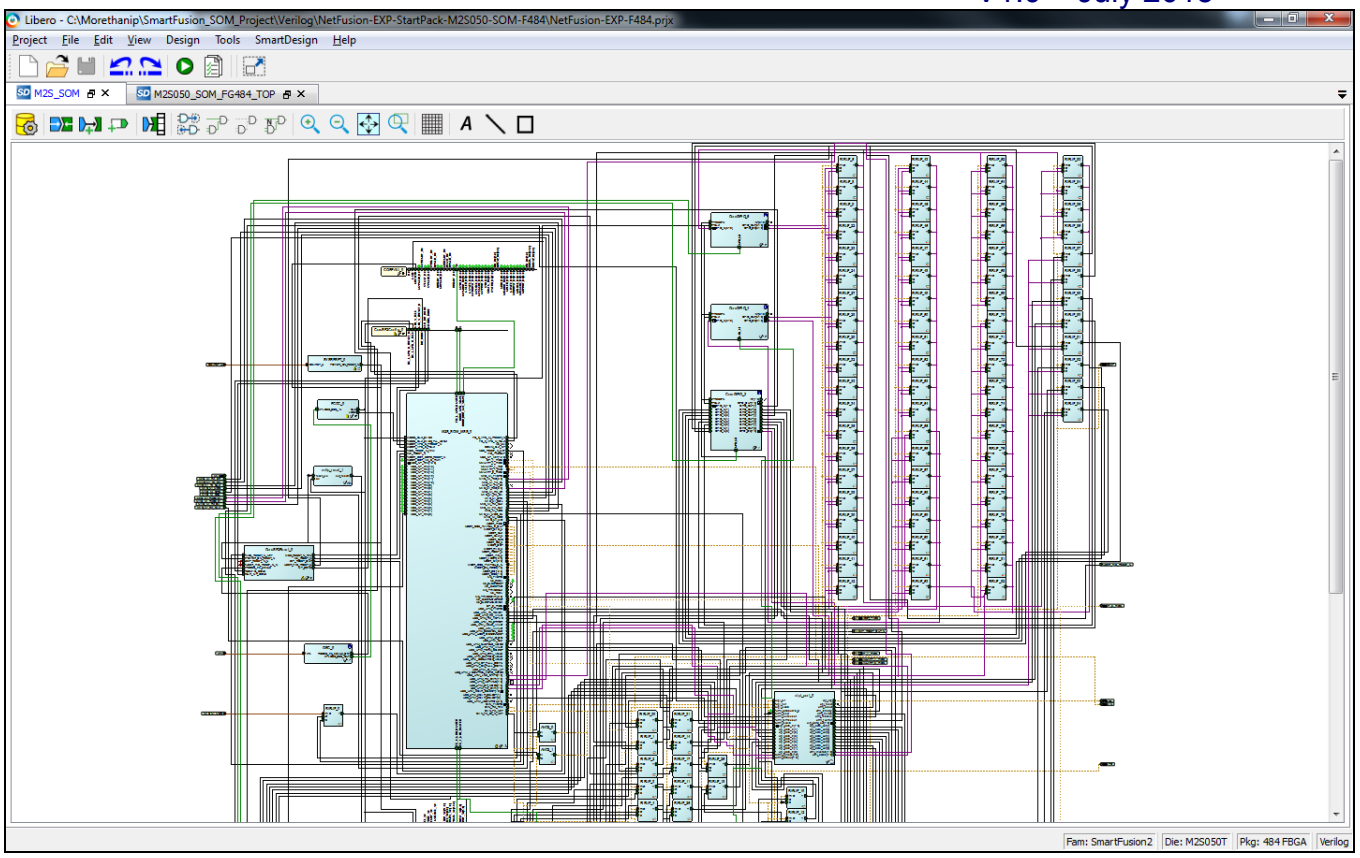


Figure 28 - Screenshot of Lower-Level SOM sheet of the Starter-Pack Libero Project

13.1 Emcraft Systems

The (*above*) figure overview encapsulates the whole of the NetFusion-EXP SOM sheet. However, it was built on the basic project supplied currently by Emcraft. They designed and developed the M2S-SOM-F484 that is housed on the NetFusion-EXP baseboard PCB.

Various IP cores were added to accommodate the need to support and facilitate the vast multitude of NetFusion-EXP's PCB hardware available to the ARM MSS sub-system in the SmartFusion2 FPGA on the Emcraft M2S-SOM-F484 housed on NetFusion-EXP as one product. As the MSS can only drive and support some of the pins on the SOM unit, there was a requirement therefore, to add more IP cores in the fabric to interface through the I/O assignments to the PCB hardware not connected directly to the MSS.

These additional cores and the default statutory parts of the SOM design in the Fabric are described in the sub-sections (*below).*

13.2 ULPI/UTMI OTG USB

On the SmartFusion2 F484 package that is used with Emcraft's M2S-SOM-F484 System-On-Module, the ULPI MSS interface is not supported. However, the UTMI OTG USB signals are supported. As the track routing and the IC USB device on the NetFusion-EXP PCB support ULPI, an IP core in the fabric is require to convert between the two different USB On-The-Go protocols and signals. This has been instantiated as [*ulpi_port_0*].

This OTG core was used from OpenCores at http://www.opencores.com and resides in the NetFusion-EXP starter project's ***hdl/*** sub-directory.

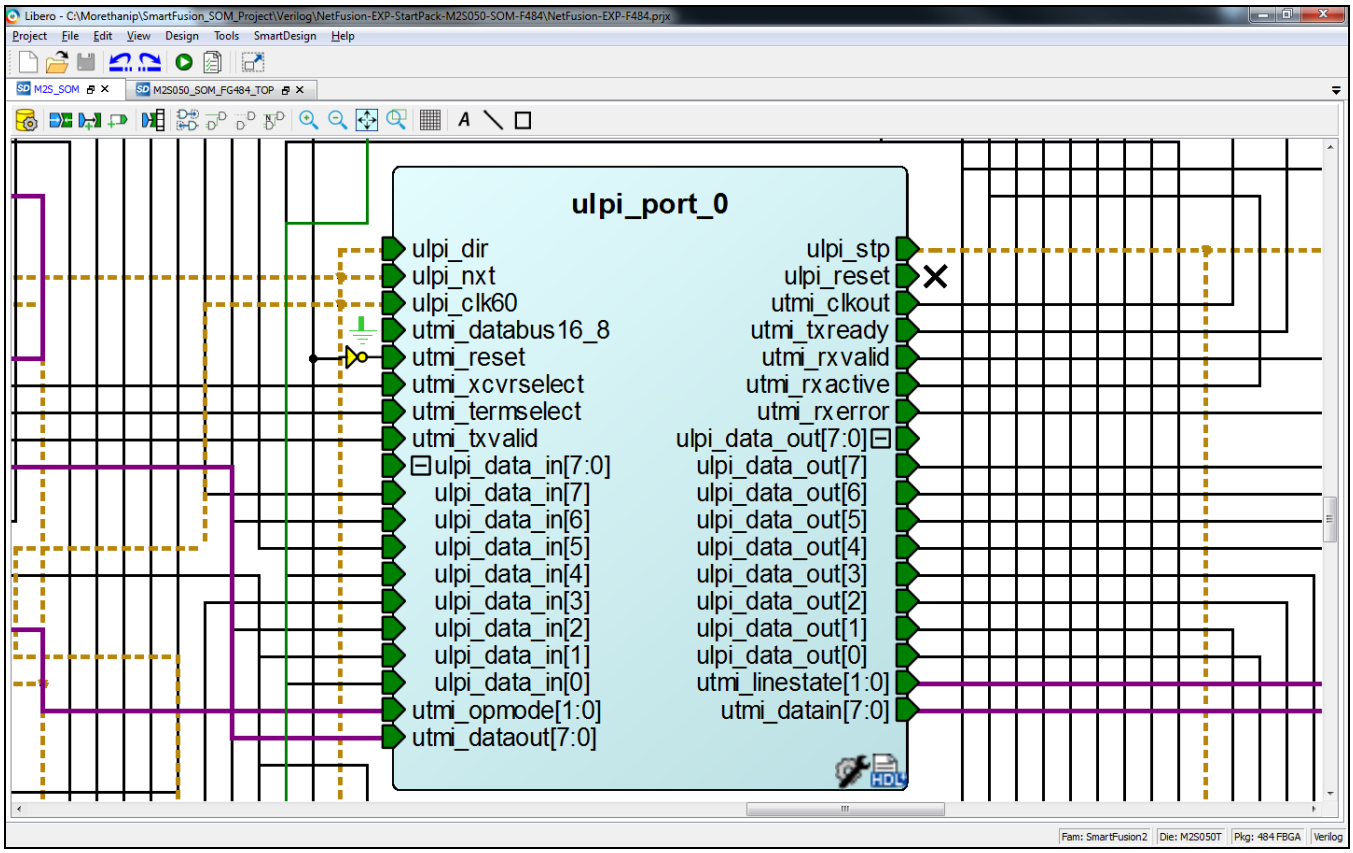


Figure 29 - ULPI/UTMI USB Converter IP Core

13.3 CoreSF2Reset

This core is used to co-ordinate the reset across all of the FPGA in strict timed sequence. It is important for the peripheral logic to be released after the MSS ARM processor and then the fabric logic following the MSS. This is a standard Libero ACTEL core from the vault.

The **EXT_RESET_OUT** signal is routed out of the FPGA device to the NetFusion-EXP PCB. The input to this core is **the POWER_ON_RESET_N**. This core was instantiated as [*CoreSF2Reset_0*].

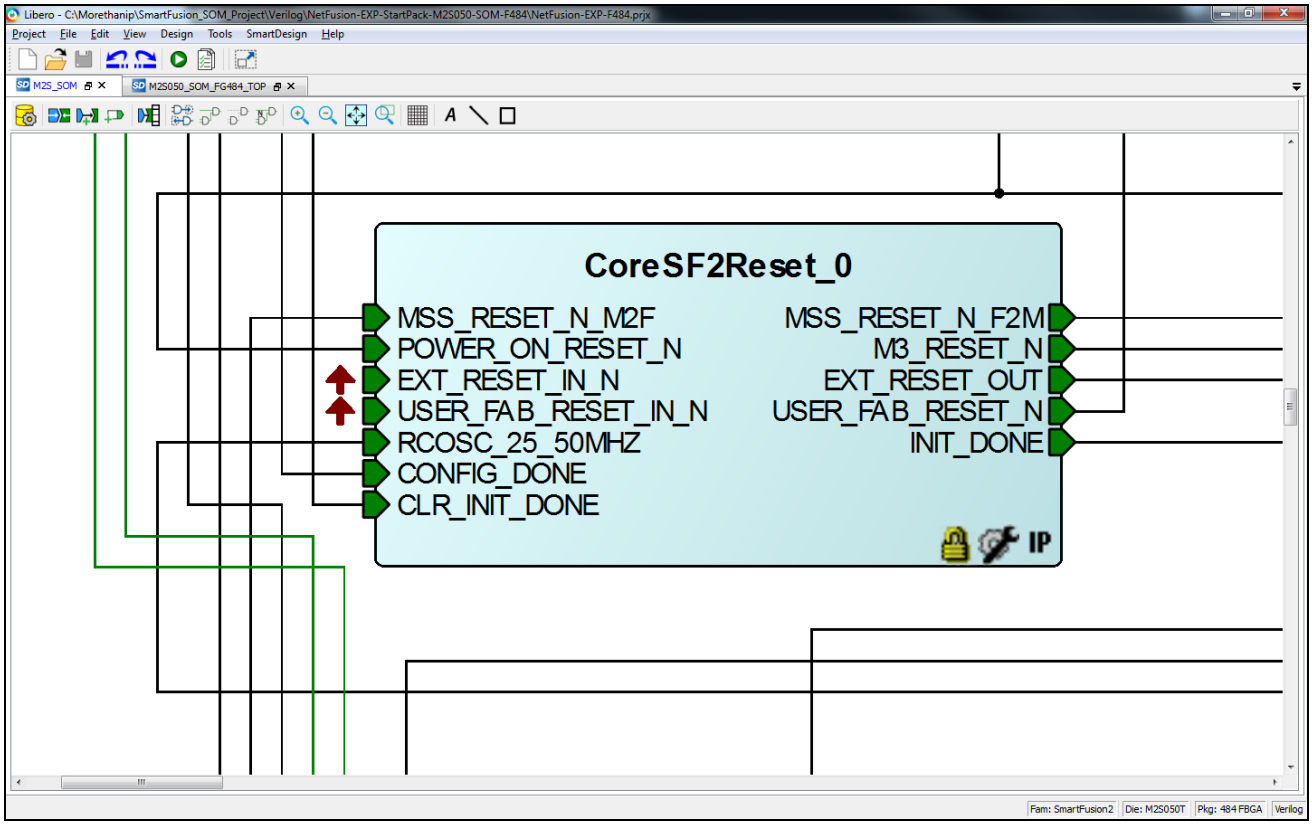


Figure 30 - SmartFusion2 Reset Controller IP Core

13.4 FCCC

The entire FPGA sub-system comprising of the MSS ARM processor and the main fabric run on clocks all generated from this core. It is the main coordination of the clock lines that are distributed. The input is *12MHz* from an off-chip crystal/IC. The CCC PLL divides down the *12MHz* source by **12** to *1MHz*. Then this is multiplied up by differing amounts for **GL0**. You can add more clock PLL lines as you wish when you are modifying the design.

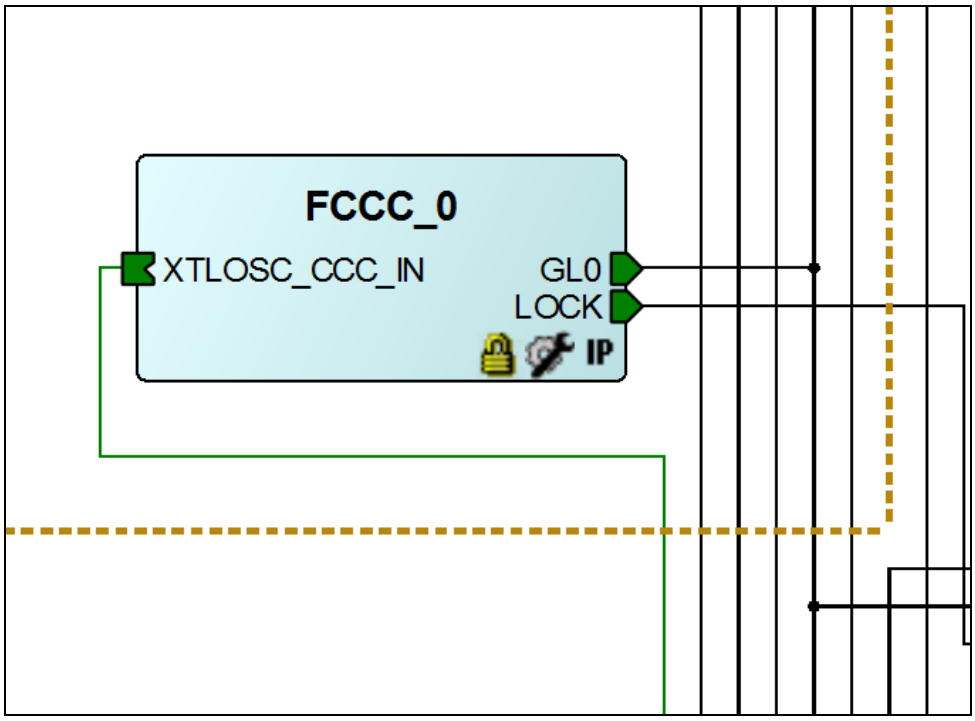


Figure 31 - Clock PLL Macro Core

The core as been instantiated as [F*CCC_0*]. There is also a **LOCK** output signal that is used by the MSS to determine when the PLL has settled and locked onto the desired output frequencies. All clock outputs are digital square waves.

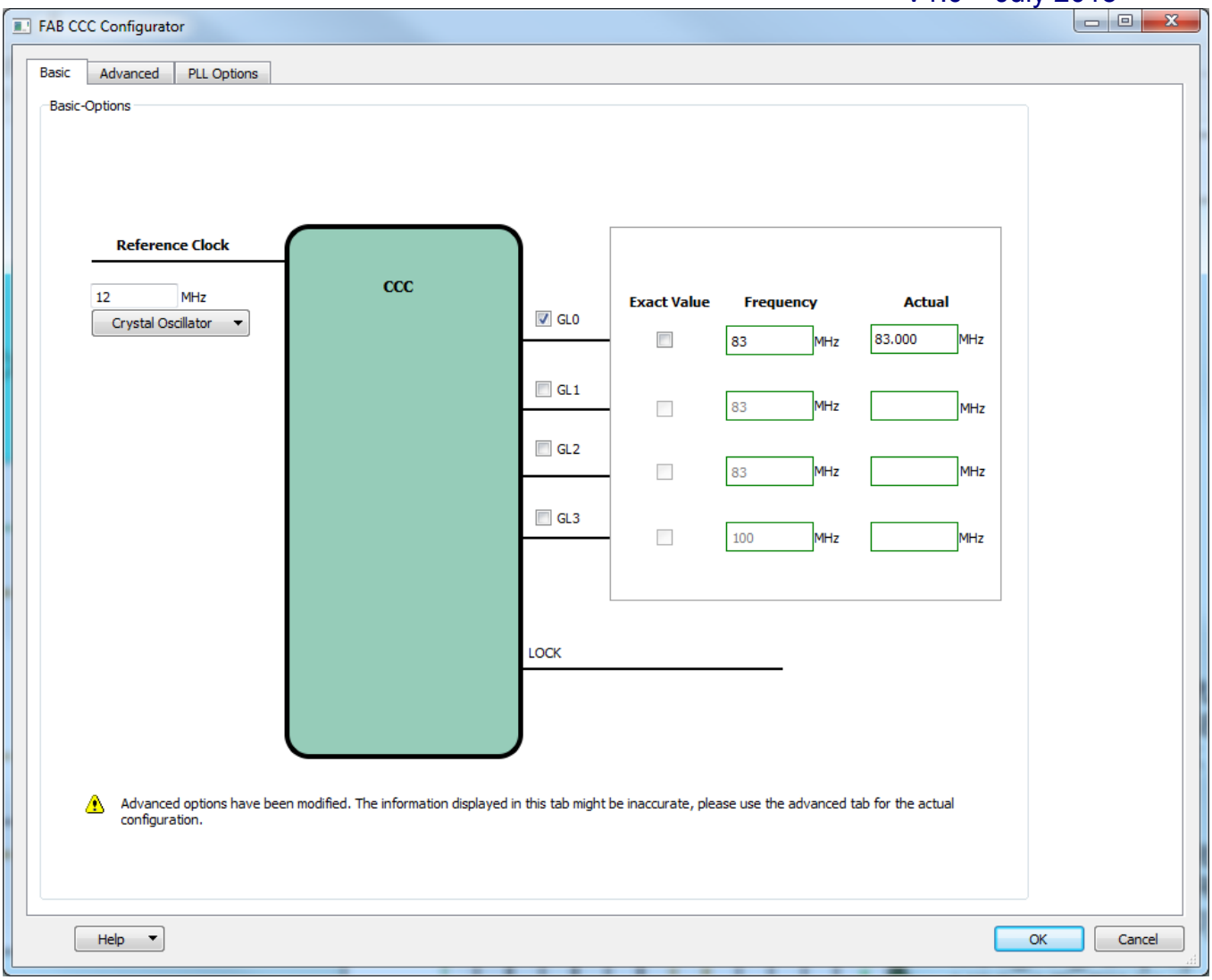


Figure 32 - PLL Clock Macro Settings

Output signal **GL0** is distributed to the memory bus and the MSS ARM block. **T**he main base clock frequency for the *166MHz* ARM processor is scaled up inside the MSS block.

13.5 Counter - CoreGPIO

NetFusion-EXP PCB has an on-board temperature sensor. The output of which is a square wave signal that has a frequency is equivalent to degrees Kelvin (*down to absolute zero*). This signal is routed into the SmartFusion2 FPGA fabric and clocks the *16-bit* counter [*counter16_0*]. The counter's *16-bit* output value is wired into GPIO input [*coreGPIO_1*] which is memory addressable from the ARM uClinux applications. This serves as a simple *32-bit* memory location to read and makes the software algorithm for determining the temperature incredibly simple.

Note: *the memory interface from the ARM MSS is an APB interface.*

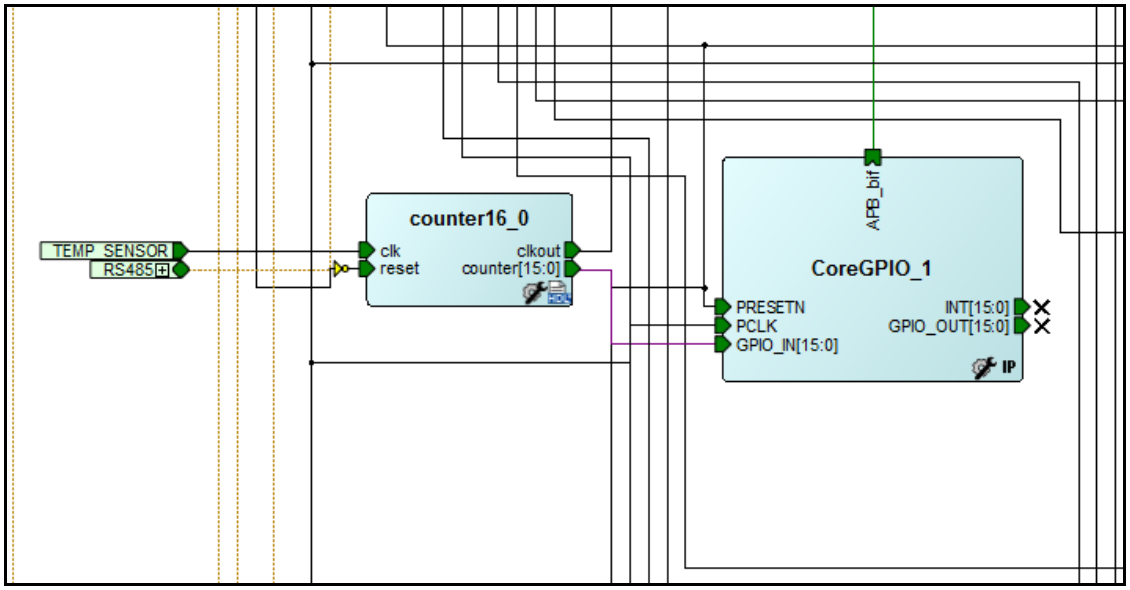


Figure 33 - Temperature Sensor Logic

13.6 NetFusion-EXP Baseboard Peripheral CoreGPIO

The vast majority of the hardware on the NetFusion-EXP PCB is connected through this core. The obvious exceptions are the Ethernet pathways, *RS485 UART* and the *SPI* audio input/output. However, all of the *I2C, Relays, I/O expanders, Real-Time clock IC, voltage monitor, expansion I/O* and all other slow speed communications are handled through this [*coreGPIO_0*] instantiated block.

It is visible to the ARM MSS processor via an APB interface memory bus. Lines can be configured as inputs or outputs and are all individually controlled and observed by the uClinux UIO device driver.

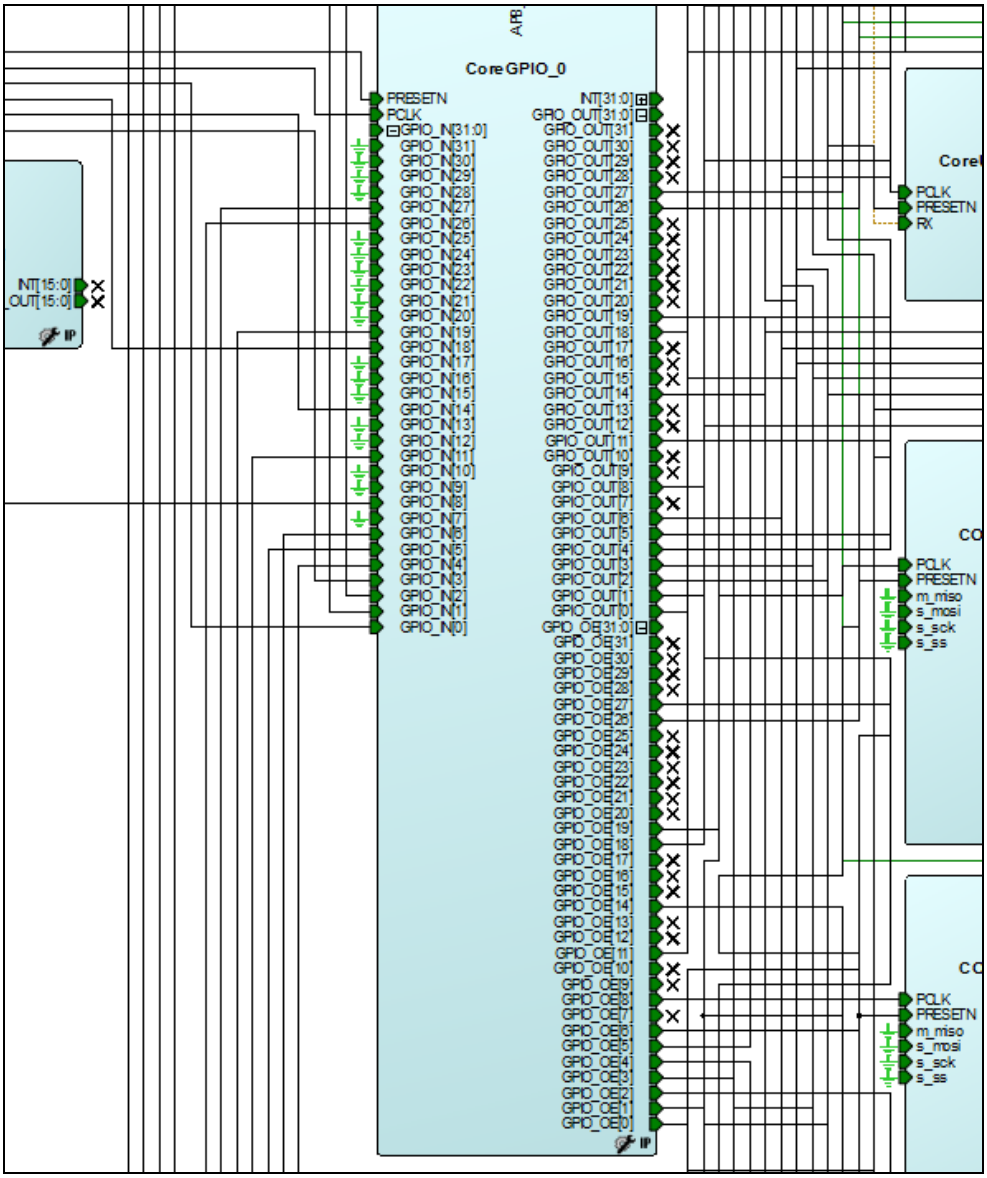


Figure 34 - Main PCB GPIO Interface with APB bus from the ARM Processor MSS block

13.7 NetFusion-EXP IO-EXP Daughter Card GPIO

The IO-EXP Daughter Card expansion slots on the NetFusion-EXP PCB are connected through these cores. The expansion slot functionality is handled through [*coreGPIO_3, coreGPIO_4, coreGPIO_5*] instantiated block and as a basic example for the user and all 72 GPIOs for the expansion daughter-card are connected to these cores for the ARM processor software to handle. It will in most cases of user development, be changed from this processor visible design to a bespoke customer logic design, but the initial design in this provided Libero project allows simple initial software testing and probing before major design changes are implemented.

The CoreGPIO blocks are visible to the ARM MSS processor via an APB interface memory bus. Lines can be configured as inputs or outputs and are all individually controlled and observed by the uClinux UIO device driver.

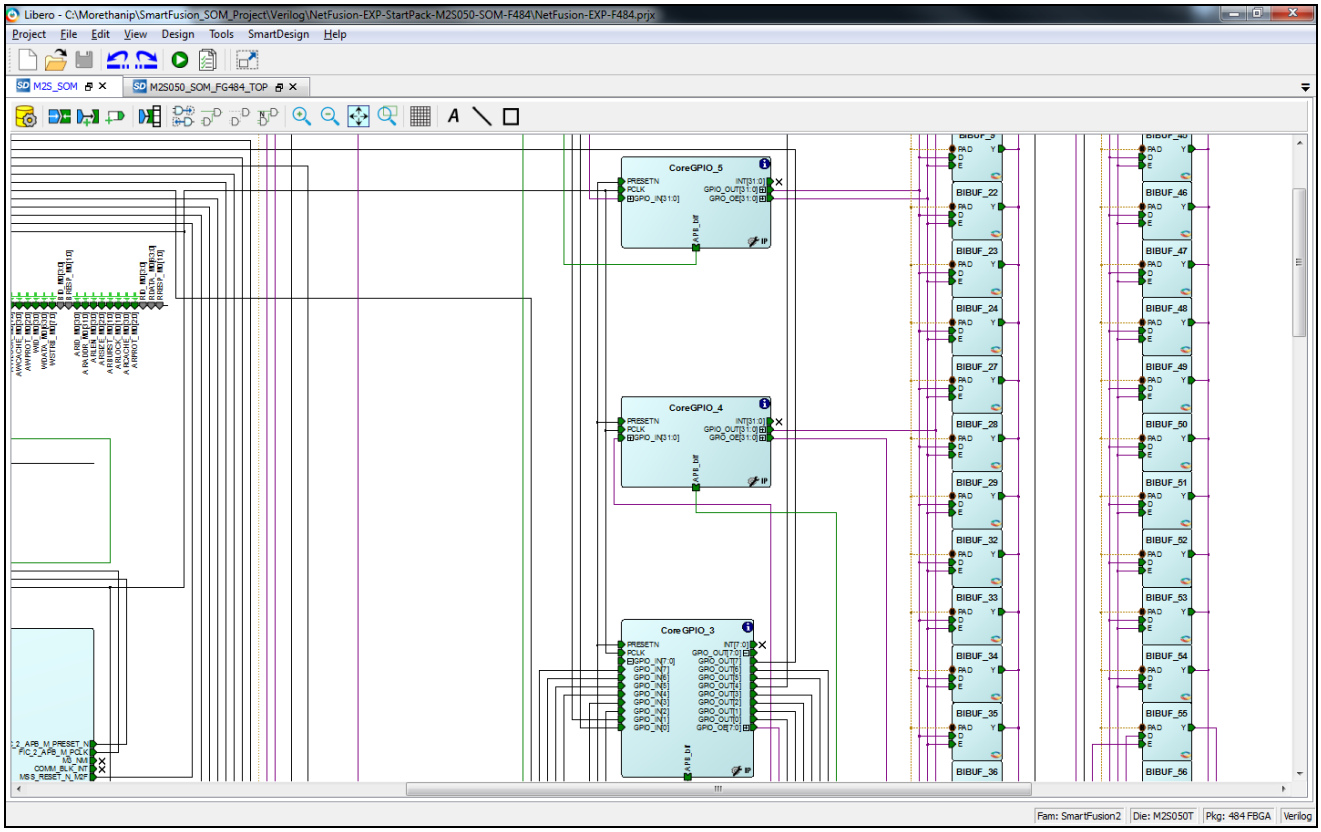


Figure 35 – Main CoreGPIO for the Daughter Card Expansion IO slots

13.8 BIBUF Bi-Direction buffers for Daughter Card GPIO

Throughout the NetFusion-EXP starter project's fabric logic design is scattered around Bi-Directional macros. However, the group shown (*below*) pertains to the bi-directional switching capability of the IO lines for ALL 72 IO-EXP daughter card signals. These convert conveniently any IN/OUT data flow when the RX and TX are kept separate in the IP cores. The output side always routes up out of the SmartFusion2 FPGA to the NetFusion-EXP daughter card slots where the signals are capable of input/output operation using pull-up resistors combining tri-state operation.

The instantiated BIBUFs are illustrated in the Figure (*below).*

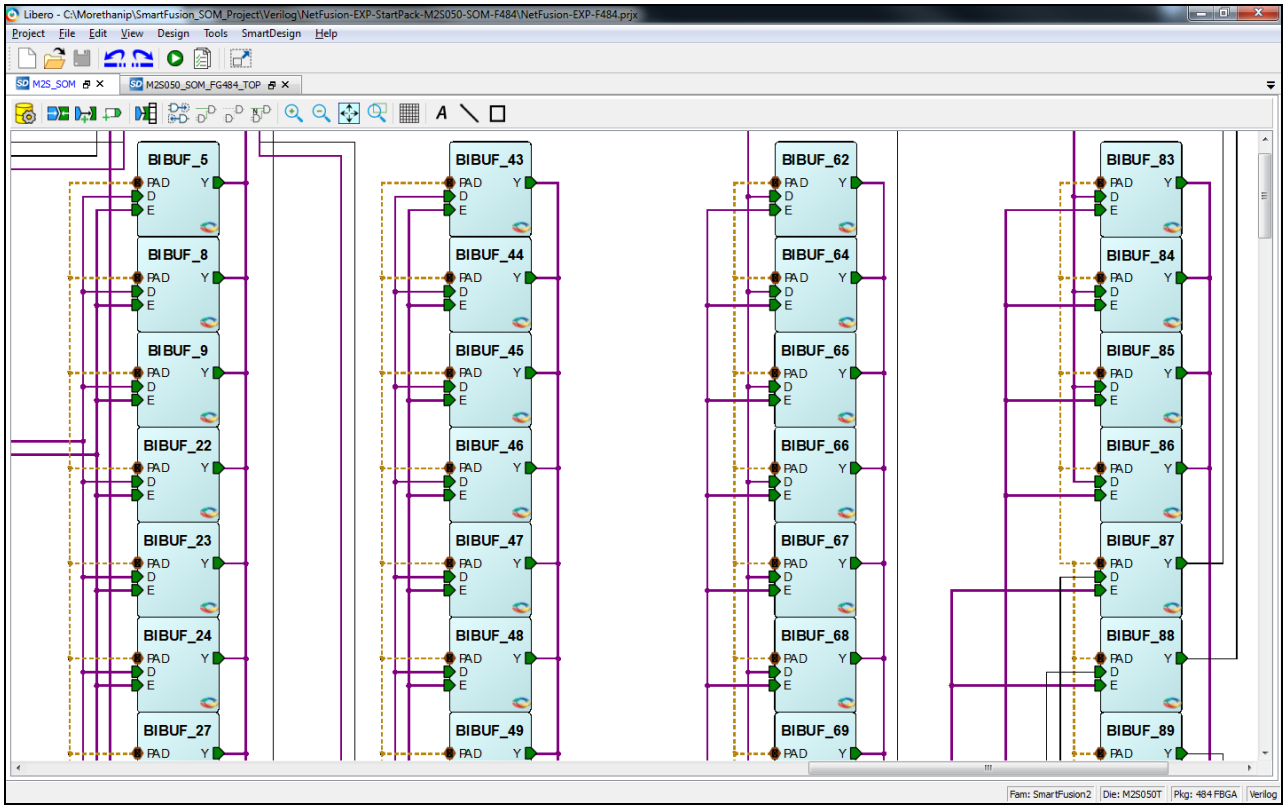


Figure 36 – Bi-Directional Buffers for the Daughter Card Expansion IO slots

Important Note: *the Libero logic design of this NetFusion-EXP start project is capable of ALL 72 IO lines to the daughter card slots to support input or output configuration. As the user customizes and changes the design this may become more specific to input or output depending the required design. Additionally, some daughter card's themselves may have input only or output only circuitry present which means if the user is keeping this starter project design, they will still have to configure the associated GPIO core to reflect the I/O setup.*

13.9 CAN Controller

The CAN transceiver on the NetFusion-EXP baseboard PCB is connected to the CAN Controller (*inside the MSS block*) either directly for the RX and TX, or via OR gate logic for the enable lines. These are instantiated as [*OR2_0, OR2_1*] and serve as a mask to either allow *bus1*, *bus2* or both to be used during transmission. The masks are controlled by signals from the small GPIO core [*CoreGPIO_2*]. The signals from the GPIO core to mask the enable lines ensures that application software controls the dual bus behavior for the CAN interface on NetFusion-EXP. Corresponding GPIO bits that are cleared disable that CAN bus and those which are set enable it.

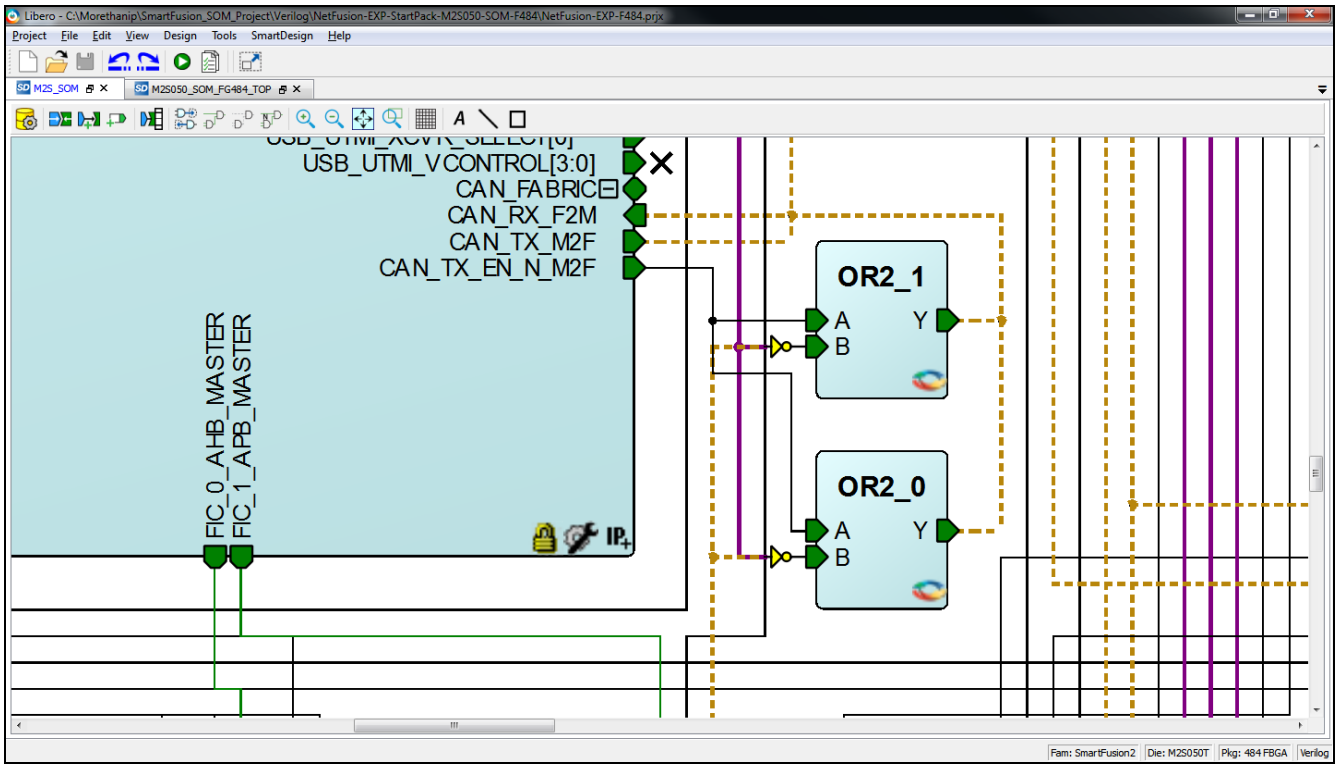


Figure 37 - CAN Controller Interface Logic

13.10 CoreUARTapb

As there is no way in the SmartFusion2 ARM MSS block for standard UART connections *(/dev/ttyS0* and */dev/ttyS1* already used) the RS484 PCB hardware has to be controlled and handled from a fabric UART core. This was instantiated as [*CoreUARTabp_0*] and has an APB connection to the MSS block.

The uClinux device driver for this hardware access the core as a block of memory and the FIFO RX and TX data is stored in the fabric core. The RX ad TX signals to the NetFusion-EXP PCB hardware are TTL levels and then get converted to RS485 voltage signals in the electronics.

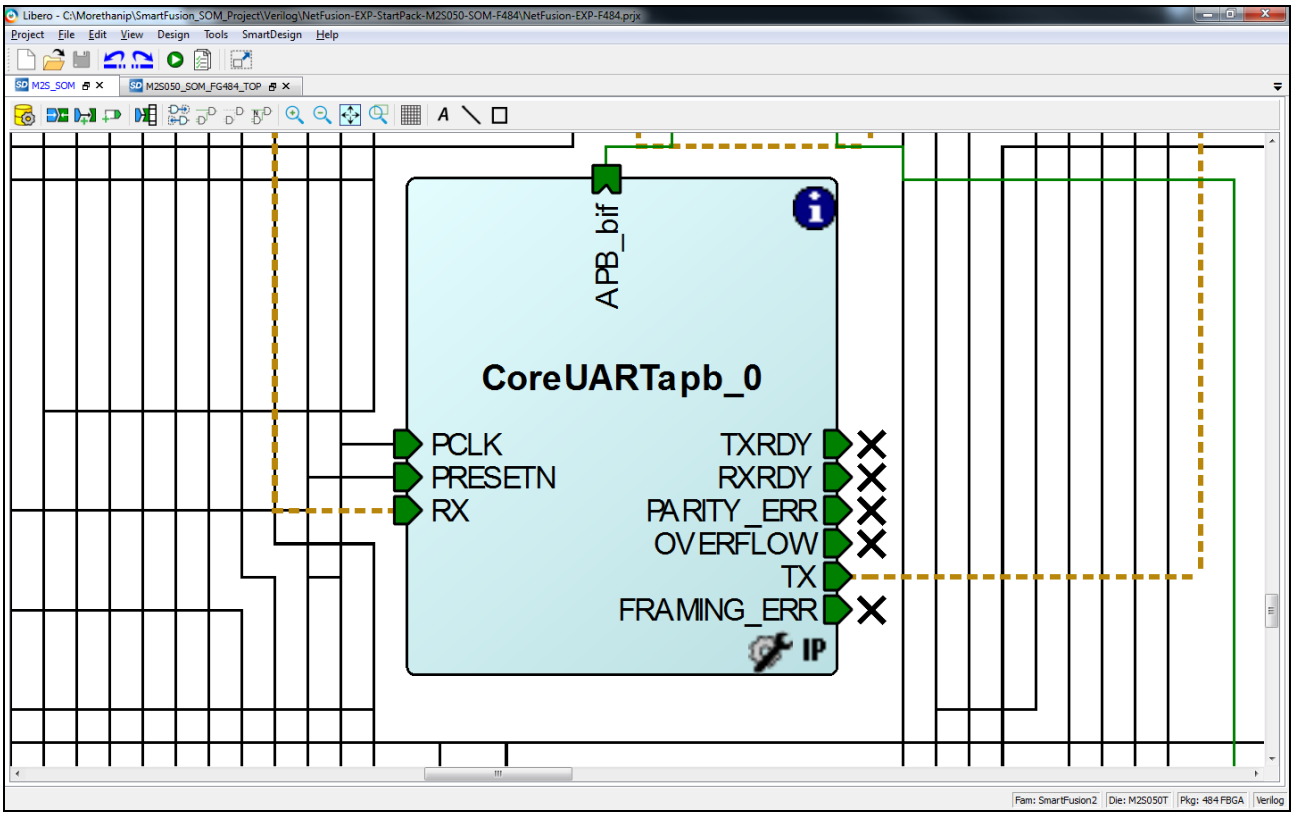


Figure 38 - RS485 UART IP core

13.11 CoreSPI0, 1

Running as separated IP cores instantiated in the fabric to achieve higher speed sampling by SPI software are SPI blocks. These are [*CORESPI_0* and *CORESPI_1*] and use an APB interface to the ARM MSS sub-system. Each core has a *4-wire* SPI bus routed out of the FPGA to the wider NetFusion-EXP PCB hardware.

These SPI interfaces connect to DAC and ADC stereo IC devices. This allows for audio to be sent and received from the PCB and the digital samples can be processed by the ARM processor from network traffic if necessary. SPI is used as it is full duplex and runs at over *1MHz* during operation. However, bottlenecks in the processor application code and also human audible hearing limitations keep realistic sampling operations around < *10KHz*.

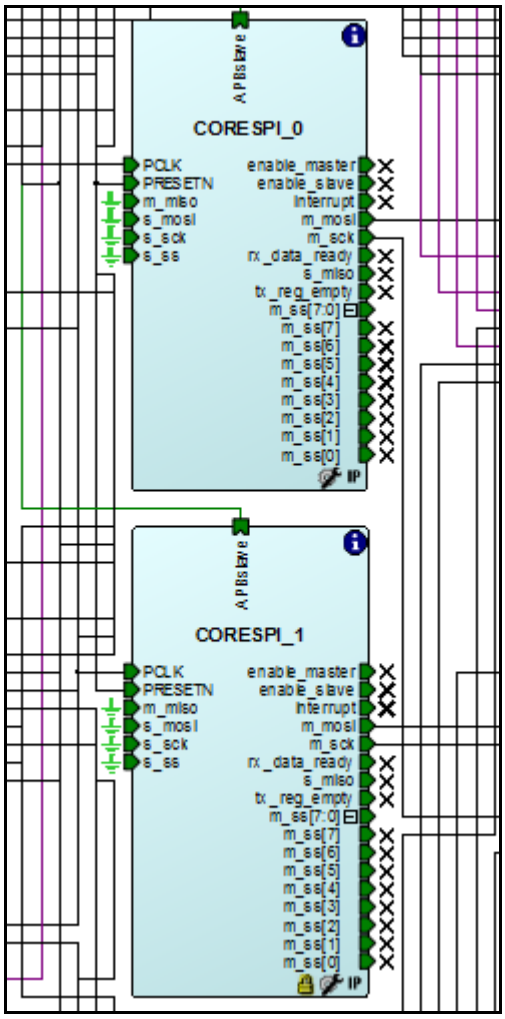


Figure 39 - Stereo Audio Line IN & Line OUT

13.12 Temperature Sensor Glue Logic

The baseboard temperature sensor IC is mounted under the Emcraft **SOM** daughter card and monitors the temperature of the FPGA that can often be demanded to perform high switching tasks in the fabric of the device.

The sensor produces an output square wave in which the frequency in *Hz* is equal to the degrees Kelvin on the baseboard. This is routed into the FPGA and the fabric and is represented on the SOM design sheet as "**TEMP_SENSOR**". This in turn, clocks the [*counter16_0*] which has a *16-bit* register output, and can be read into the ARM software environment via the [*CoreGPIO_1*] APB3 core.

Although this read value wraps around *16-bit* values, the software can use a one second timer to run comparisons and then calculate the temperature in Kelvin.

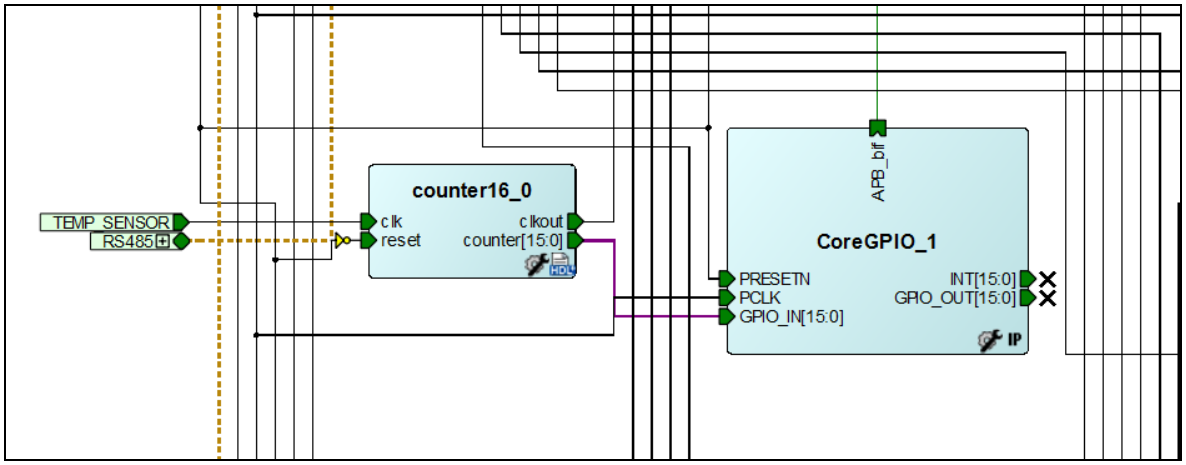


Figure 40 - Temp Sensor Counter Logic

13.13 Coordinated RESET

The customized IP core instantiated as [*mtip_reset_1*] coordinates a longer reset pulse than the power on reset core often provides. The source [*SYSRESET_0*] feeds into the [*mtip_reset_1*] which then provides an output to the whole SOM sheet design. This includes the *ARM MSS block*, *memory cores*, *GPIO* and all peripherals.

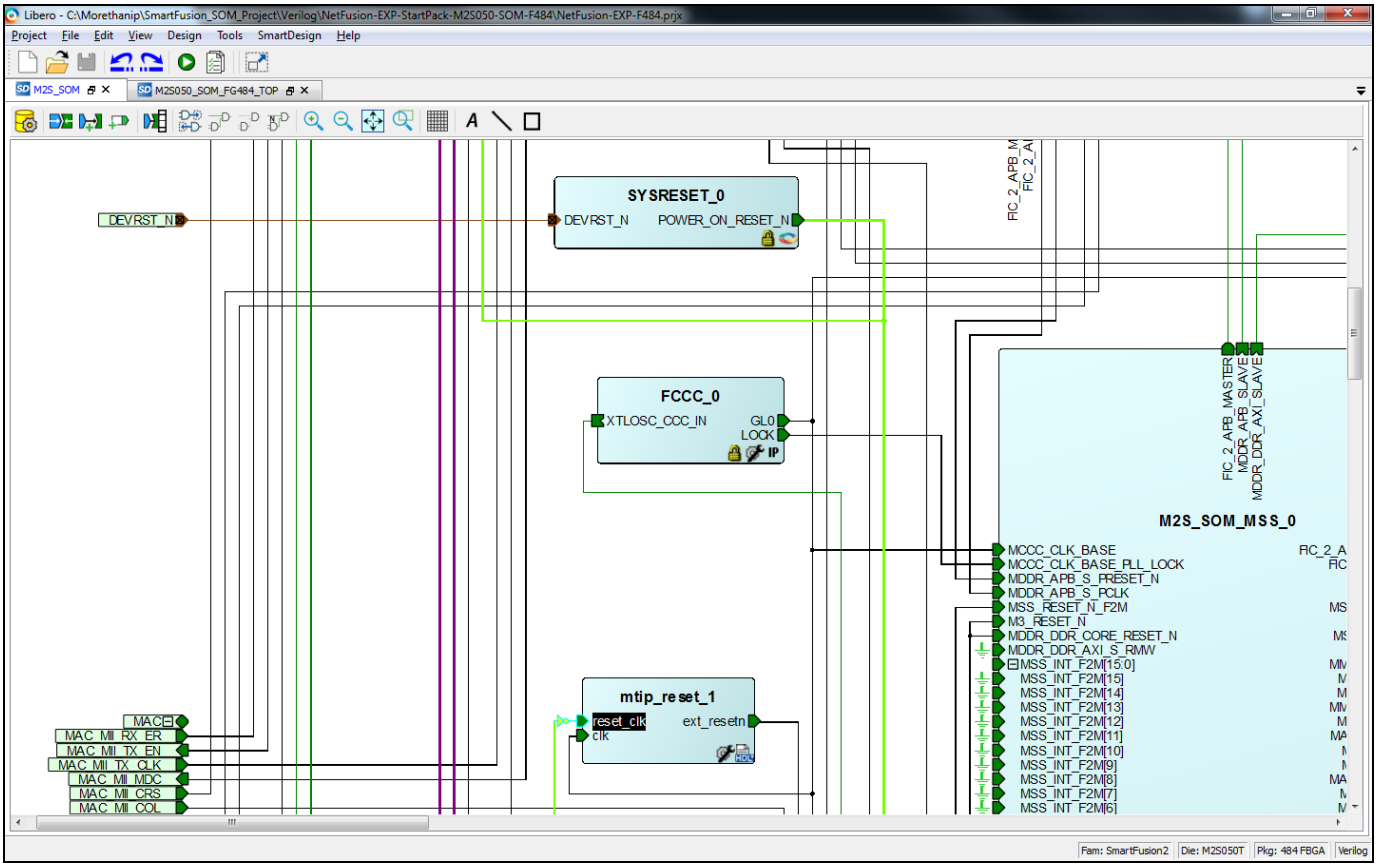


Figure 41- Customized RESET logic

13.14 BIBUF

Throughout the NetFusion-EXP starter project's fabric logic design is scattered around Bi-Directional macros. These convert conveniently any IN/OUT data flow when the RX and TX are kept separate in the IP cores. The output side always routes up out of the SmartFusion2 FPGA to the NetFusion-EXP PCB hardware where the signals are capable of input/output operation using pull-up resistors combining tri-state operation.

The instantiated BIBUFs are illustrated in the Figure (*below).*

They are controlled by an output enable signal that drives the state of the output pin.

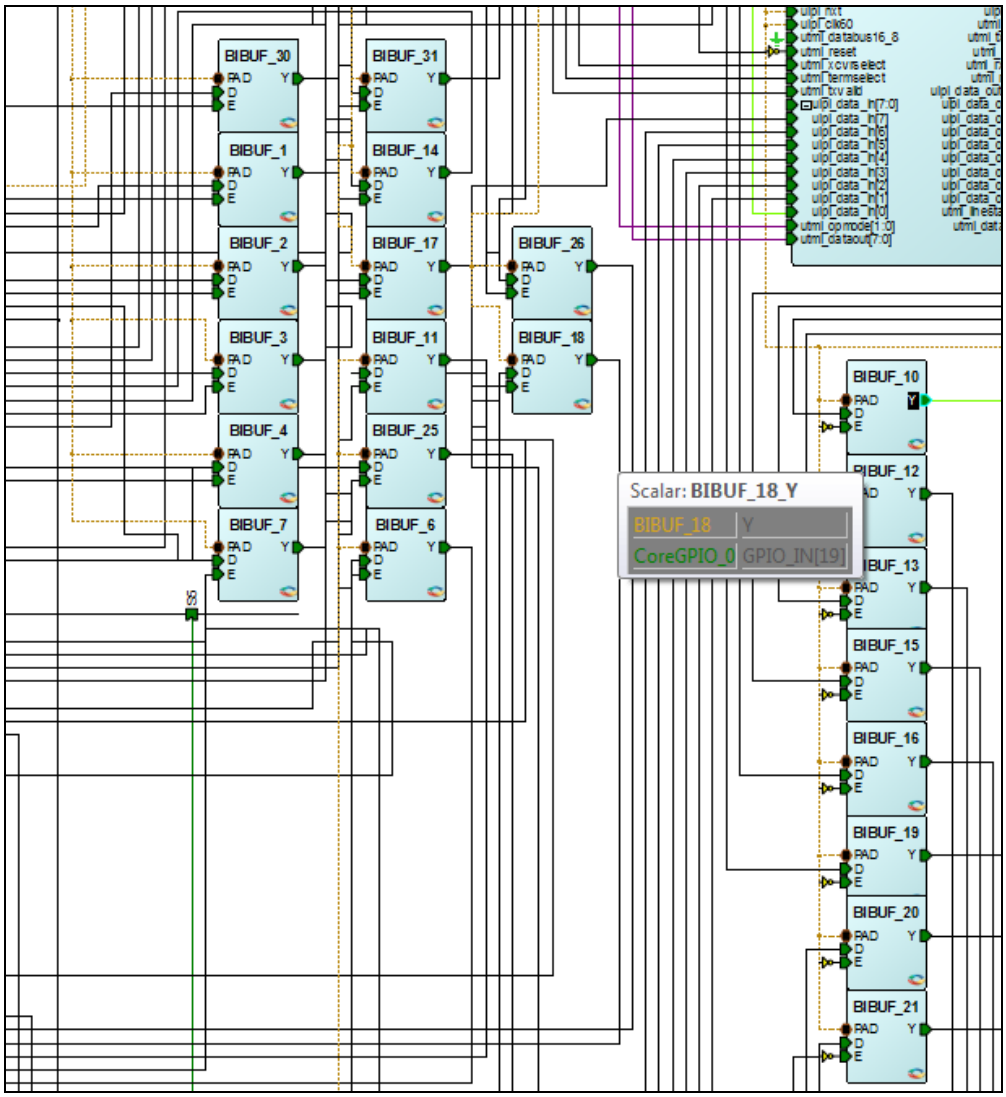


Figure 42 - Bi-Directional Fabric Macros for input/output signals

13.15 CoreSF2Config

The SmartFusion2 MSS processor sub-system always as standard has a default *CoreSF2Config* block which loops back an APB bus out and then back into the MSS block. This seems at first strange and can be very confusing. However, it is the inherent architecture of the FPGA ASIC area that most of the peripheral devices inside the MSS ARM processor core are not actually controlled by the selections made in the Libero 11.X IDE. It is the software in *u-boot* during boot-up that configures. If say for instance, an AHB-Lite interface is selected in Libero, then this does not configure the SmartFusion2 FPGA itself. It saves a configuration file that can be included by software in either bare-metal programming or the *u-boot* from Emcraft uClinux environment.

Only when the boot-up code access the APB feedback bus via [*CoreSF2Config_0*] and manipulates hardware peripheral memory address does the peripherals in the MSS get the correct mode of operation intended for them.

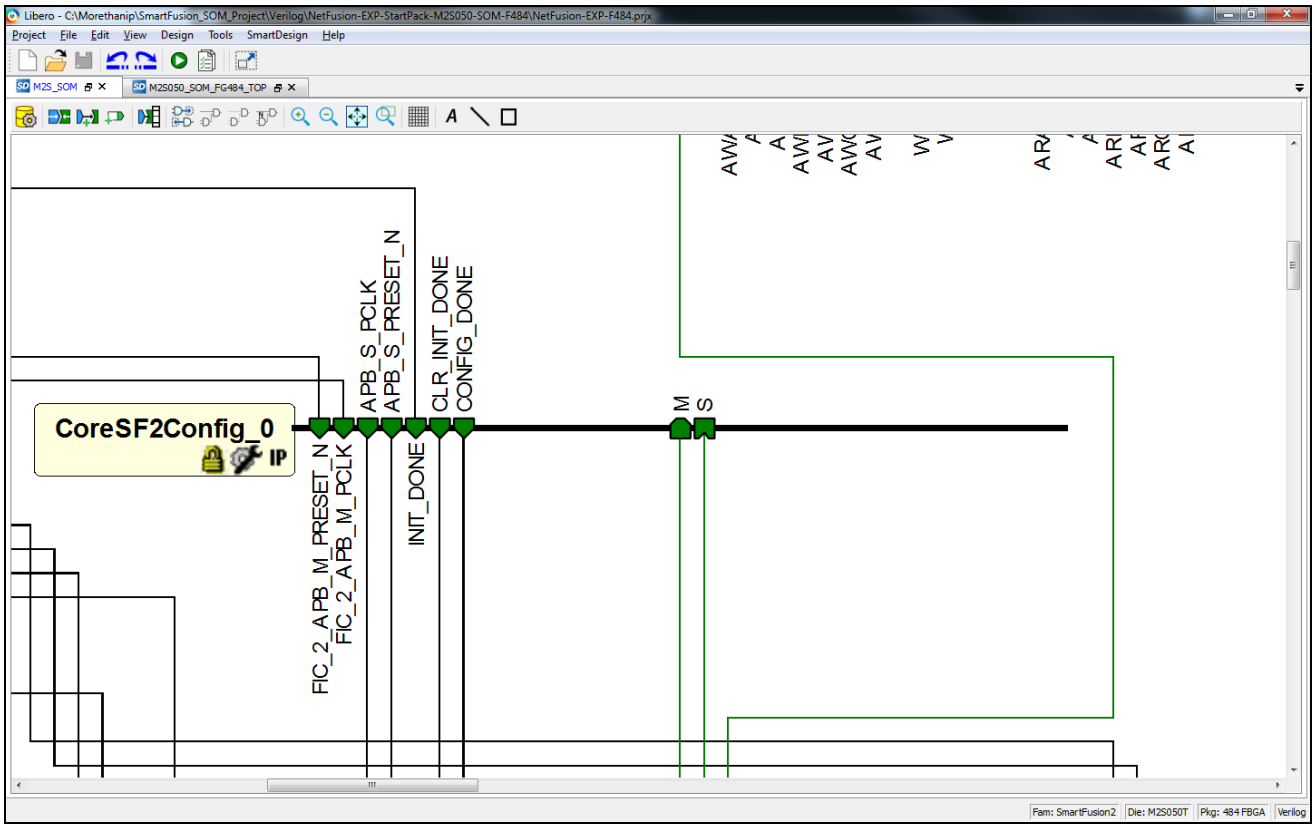


Figure 43 - APB feedback Bus for Peripheral Configuration by Software

13.16 Core AXI

In the SOM sheet of the design, there exists an IP core used for AXI memory bus accessing. It does not have any memory core attached as default when you open and view the project initially. However, this stub is the user to use to attach to any AXI memory bus interface IP core that they wish to use. All of the signals not used are either tied *LOW* (or *HIGH* if required) for input signals, and marked as unused if output signals.

The AXI core is instantiated as [*COREAXI_0]* in the SOM sheet design and currently has no intended functionality other than providing the user with a hook access point to the AXI ARM MSS system. The AXI core has an associated location in the overall ARM AHB matrix memory map and therefore attached IP cores can then have hardware address visibility from the ARM processor software environment.

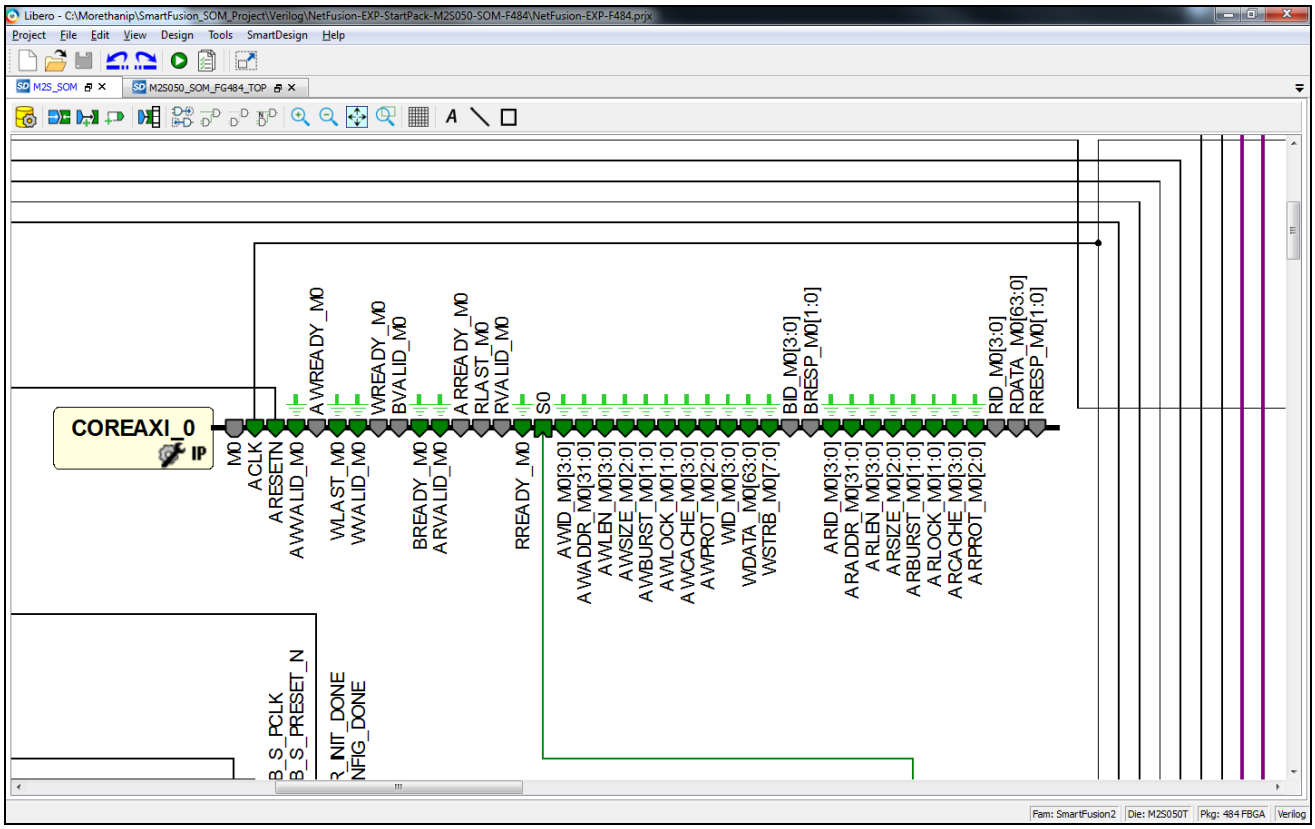


Figure 44 - AMBA Memory Interfaces from ARM for an AXI bus

13.17 Core AHB-Lite

In the SOM sheet of the design, there exists an IP core used for AHB-lite memory bus accessing. It does not have any memory core attached as default when you open and view the project initially. However, this stub is the user to use to attach to any AHB-lite memory bus interface IP core that they wish to use. All of the signals not used are either tied *LOW* (or *HIGH* if required) for input signals, and marked as unused if output signals.

The AHB core is instantiated as [*CoreAHBLite_0]* in the SOM sheet design and currently has no intended functionality other than providing the user with a hook access point to the AHB ARM MSS system. The AHB core has an associated location in the overall ARM AHB matrix memory map and therefore attached IP cores can then have hardware address visibility from the ARM processor software environment.

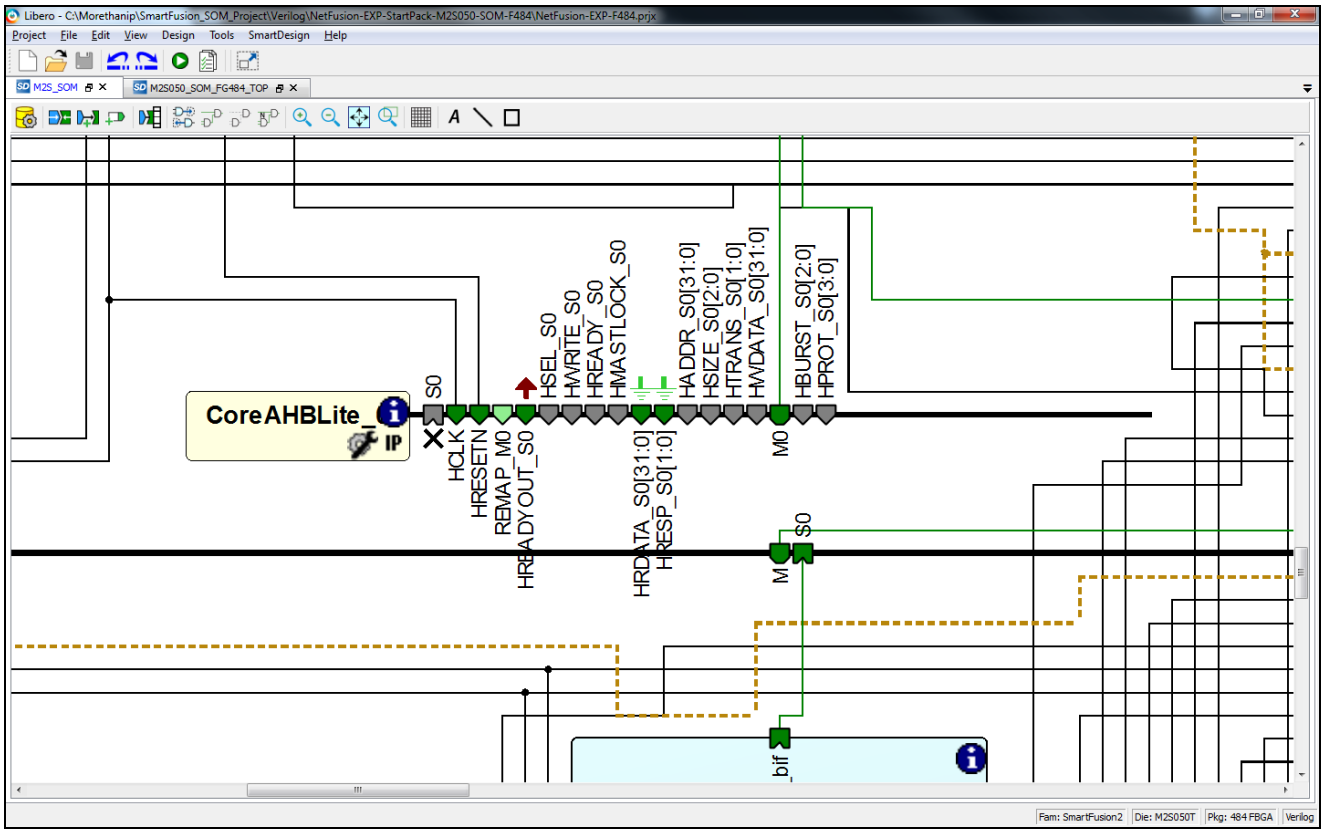


Figure 45 - AMBA Memory Interfaces from ARM for an AHB-Lite bus

13.18 Core APB3

The SOM sheet design has a useful APB3 core instantiated as [*CoreAPB3_0]* which is by default, used to collate and connect various important cores to the ARM MSS memory map. Currently the temperature sensor counter GPIO *(CoreGPIO_1),* the main GPIO core (*CoreGPIO_0*), the UART core (*CoreUARTapb_0*), the two audio SPI cores (*CORESPI_0, CORESPI_1*), the SPI baseboard device IC chip select logic (*CoreGPIO_2*) and finally the IO-Expansion GPIO for the EXP daughter-card slots (*CoreGPIO_3, CoreGPIO_4, CoreGPIO_5*).

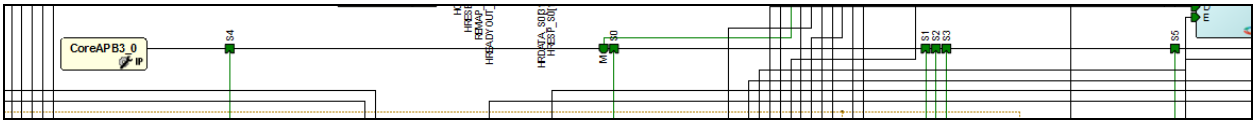


Figure 46 - APB3 fixed bus UART, GPIO, IO-EXP slots and Counter Peripherals

Although the APB3 core is used extensively, this can be extended to allow any user instantiated core to be connected by adding another slave line to the core. This can be obtained by double-clicking on the core and editing the settings.

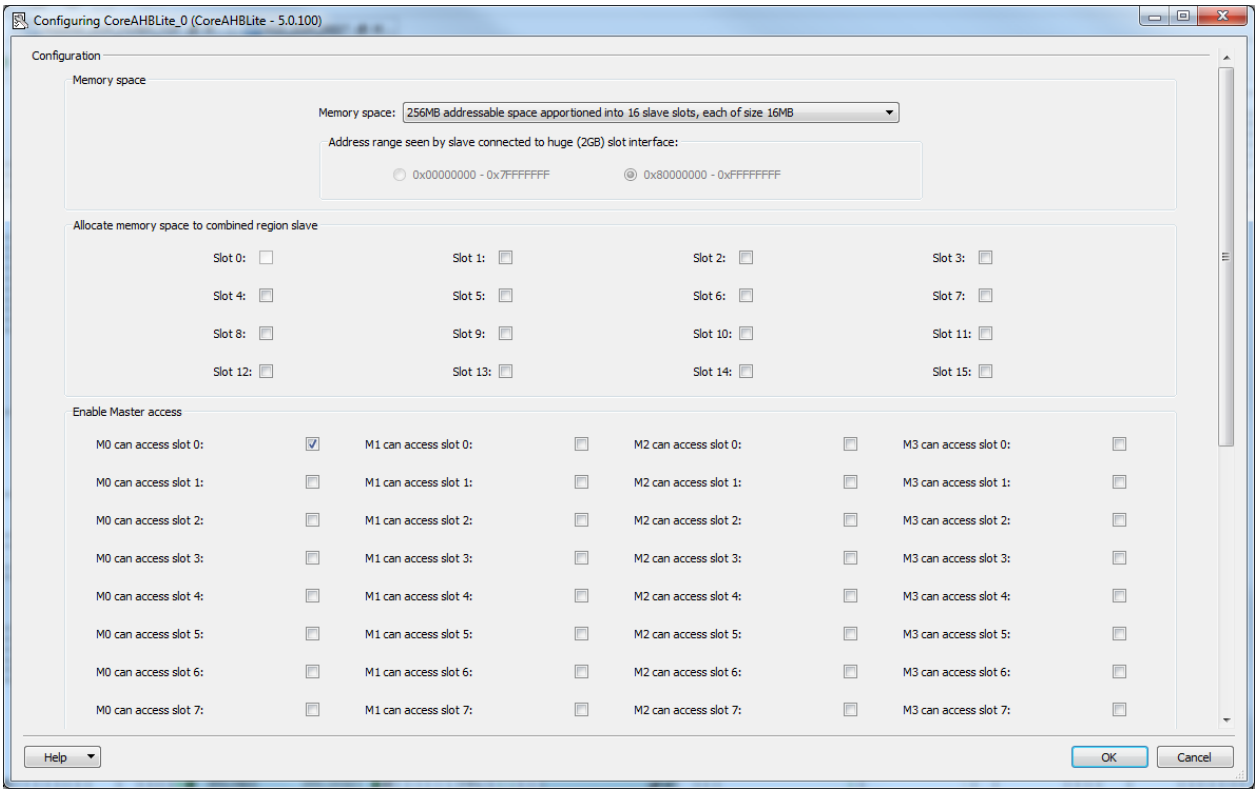

Configuring CoreAHBLite_0 (CoreAHBLite - 5.0.100)
Configuration
Memory space
Memory space: 256MB addressable space apportioned into 16 slave slots, each of size 16MB
Address range seen by slave connected to huge (2GB) slot interface:
0x00000000 - 0x7FFFFFFF
0x80000000 - 0xFFFFFFFF
Allocate memory space to combined region slave
Slot 0: Slot 1: Slot 2: Slot 3:
Slot 4: Slot 5: Slot 6: Slot 7:
Slot 8: Slot 9: Slot 10: Slot 11:
Slot 12: Slot 13: Slot 14: Slot 15:
Enable Master access
M0 can access slot 0: M1 can access slot 0: M2 can access slot 0: M3 can access slot 0:
M0 can access slot 1: M1 can access slot 1: M2 can access slot 1: M3 can access slot 1:
M0 can access slot 2: M1 can access slot 2: M2 can access slot 2: M3 can access slot 2:
M0 can access slot 3: M1 can access slot 3: M2 can access slot 3: M3 can access slot 3:
M0 can access slot 4: M1 can access slot 4: M2 can access slot 4: M3 can access slot 4:
M0 can access slot 5: M1 can access slot 5: M2 can access slot 5: M3 can access slot 5:
M0 can access slot 6: M1 can access slot 6: M2 can access slot 6: M3 can access slot 6:
M0 can access slot 7: M1 can access slot 7: M2 can access slot 7: M3 can access slot 7:
Help
OK
Cancel

Figure 47 – Editing the APB3 core settings

Please Note: *the previous AXI and AHB-Lite cores can be edited and changed with exactly the same method by double-clicking on the cores themselves.*

13.19 AXI, AHB-Lite & APB3 default Memory Mapped Settings

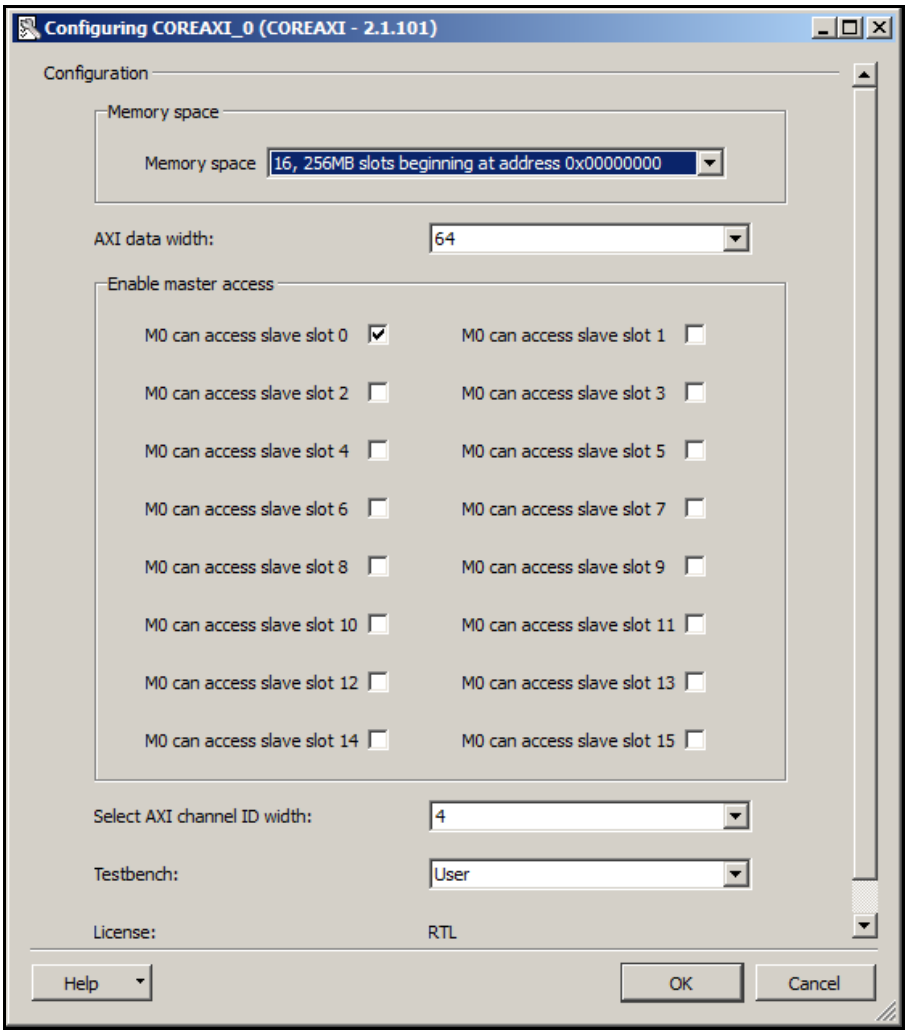

Configuring COREAXI_0 (COREAXI - 2.1.101)
Configuration
Memory space
Memory space
16, 256MB slots beginning at address 0x00000000
AXI data width:
64
Enable master access
M0 can access slave slot 0
M0 can access slave slot 1
M0 can access slave slot 2
M0 can access slave slot 3
M0 can access slave slot 4
M0 can access slave slot 5
M0 can access slave slot 6
M0 can access slave slot 7
M0 can access slave slot 8
M0 can access slave slot 9
M0 can access slave slot 10
M0 can access slave slot 11
M0 can access slave slot 12
M0 can access slave slot 13
M0 can access slave slot 14
M0 can access slave slot 15
Select AXI channel ID width:
4
Testbench:
User
License:
RTL
Help
OK
Cancel

Figure 48 - Default AXI Configuration

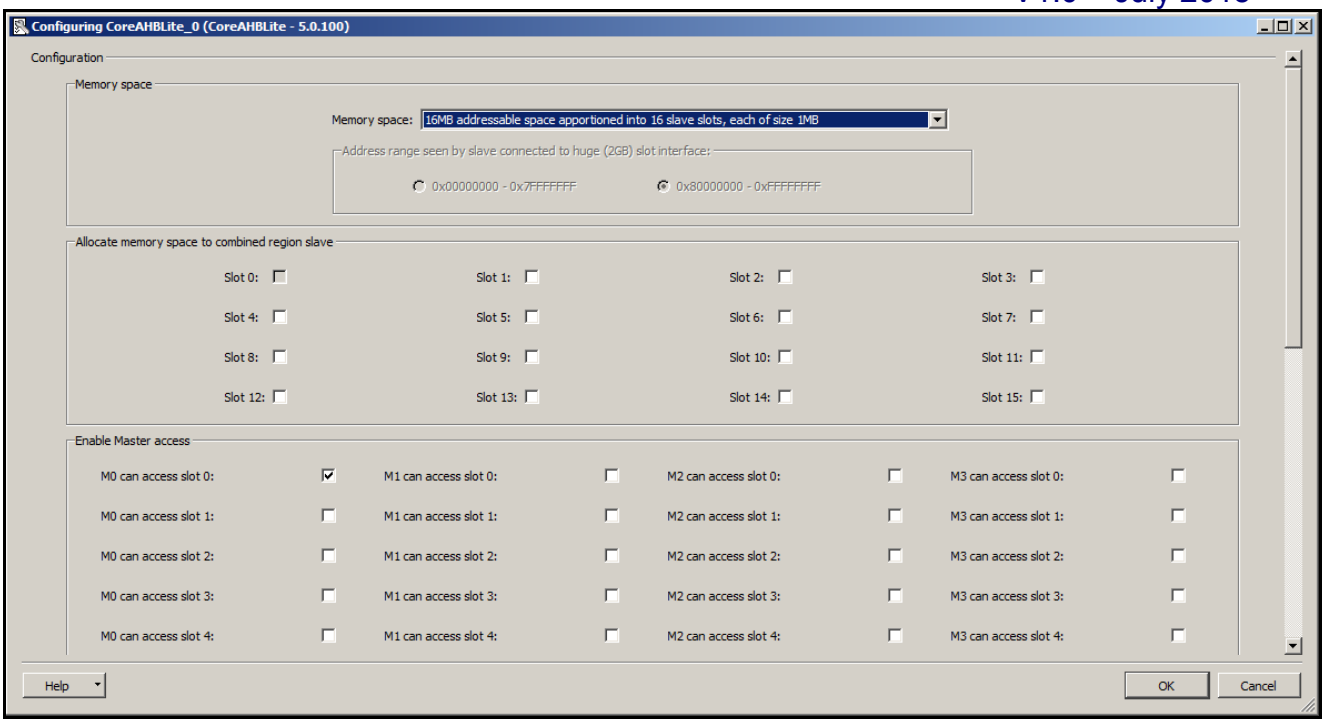


Figure 49 - Default AHB-Lite Configuration

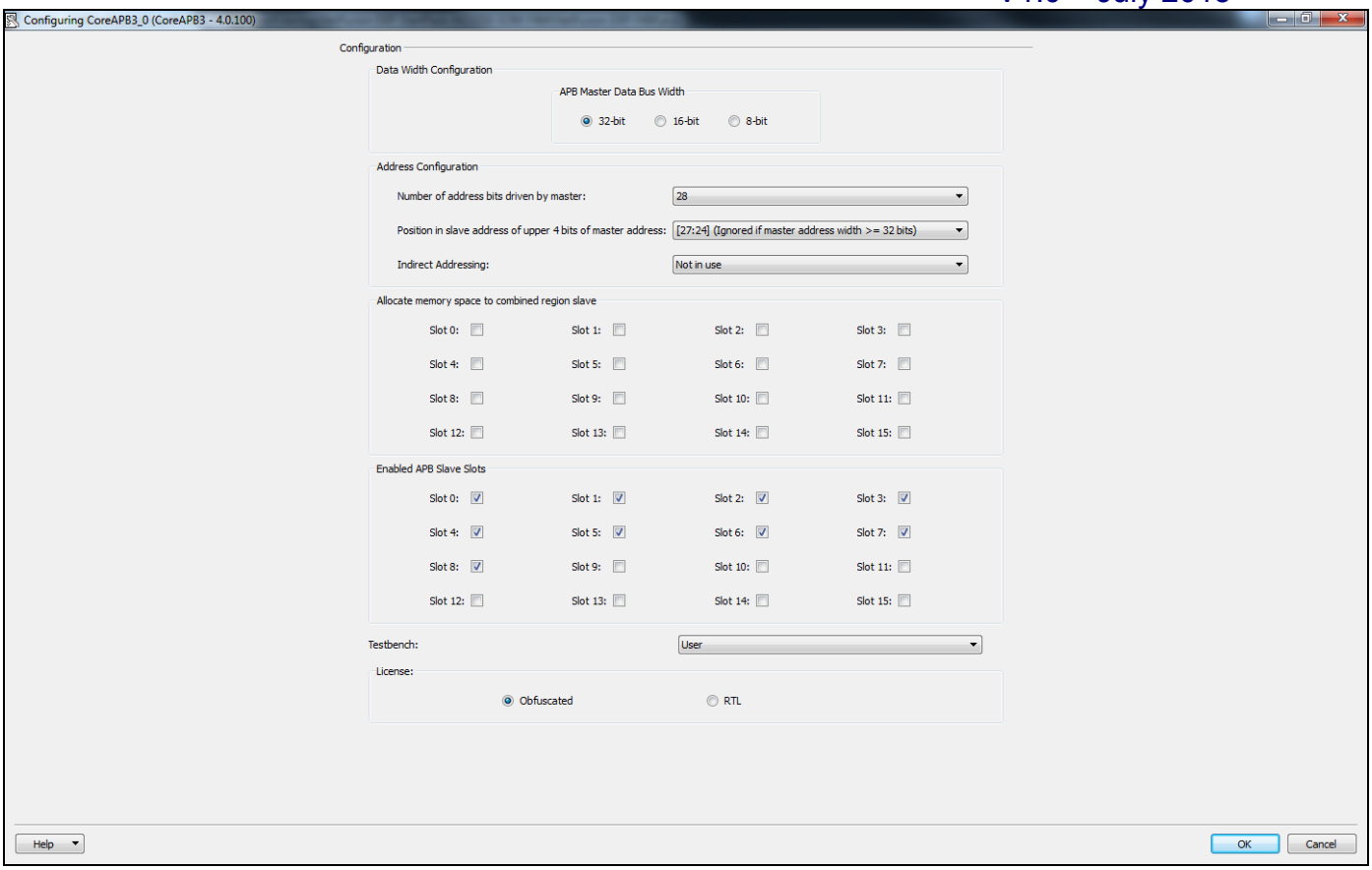


Figure 50 - Default APB Configuration

Important Note: *as a Libero 11.X designer and developer, the user may change these defaults and added or remove memory accessible IP cores to/from the fabric.*

13.20 SmartFusion2 MSS

In the SOM layer sheet of the NetFusion-EXP starter project, the MSS ARM processor main core is situated in the middle and is shown (*below*). If you double-click on the MSS block it will expanded and create another window in Libero 11.X on screen.

Note: *Refer to the next section for the MSS documentation.*

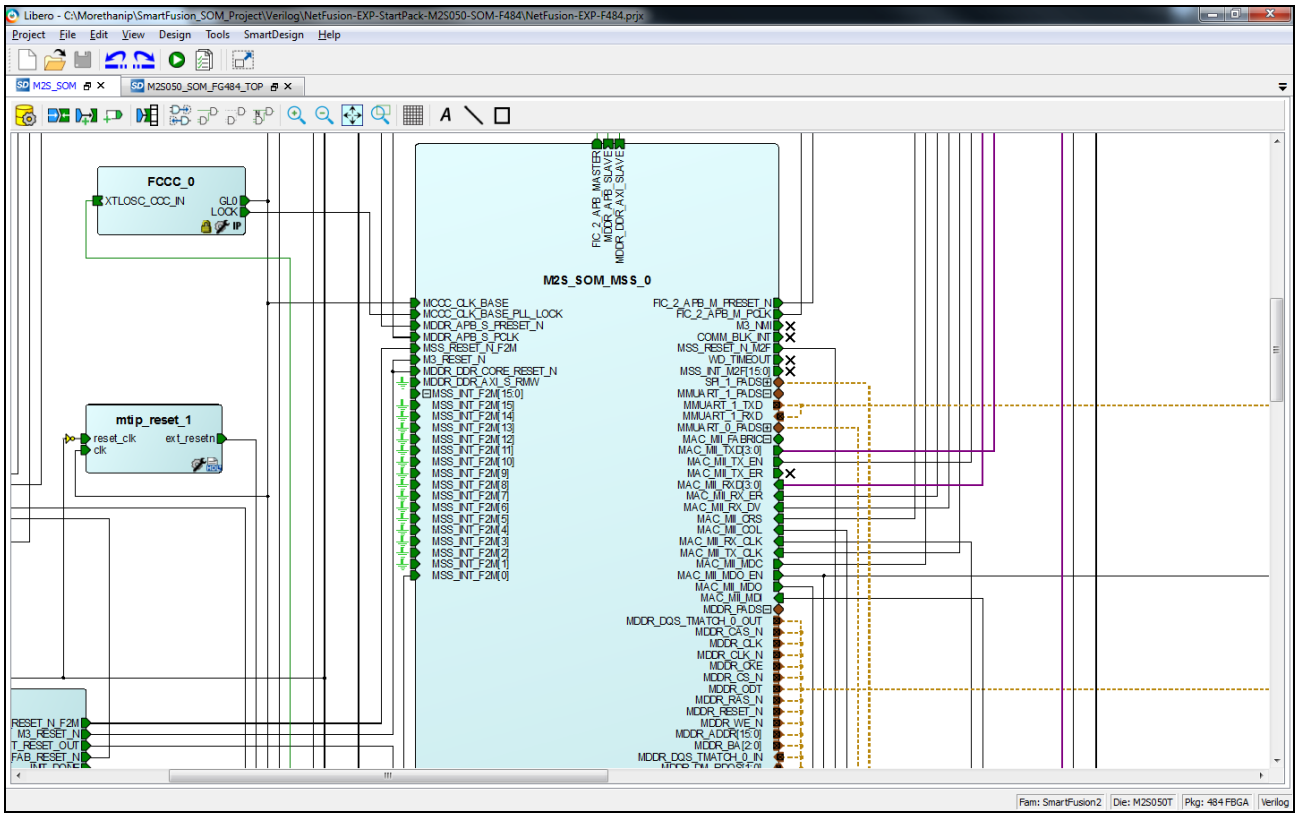


Figure 51 - SOM Module of the ARM MSS

14 NetFusion-EXP ARM Cortex MSS Default Configuration

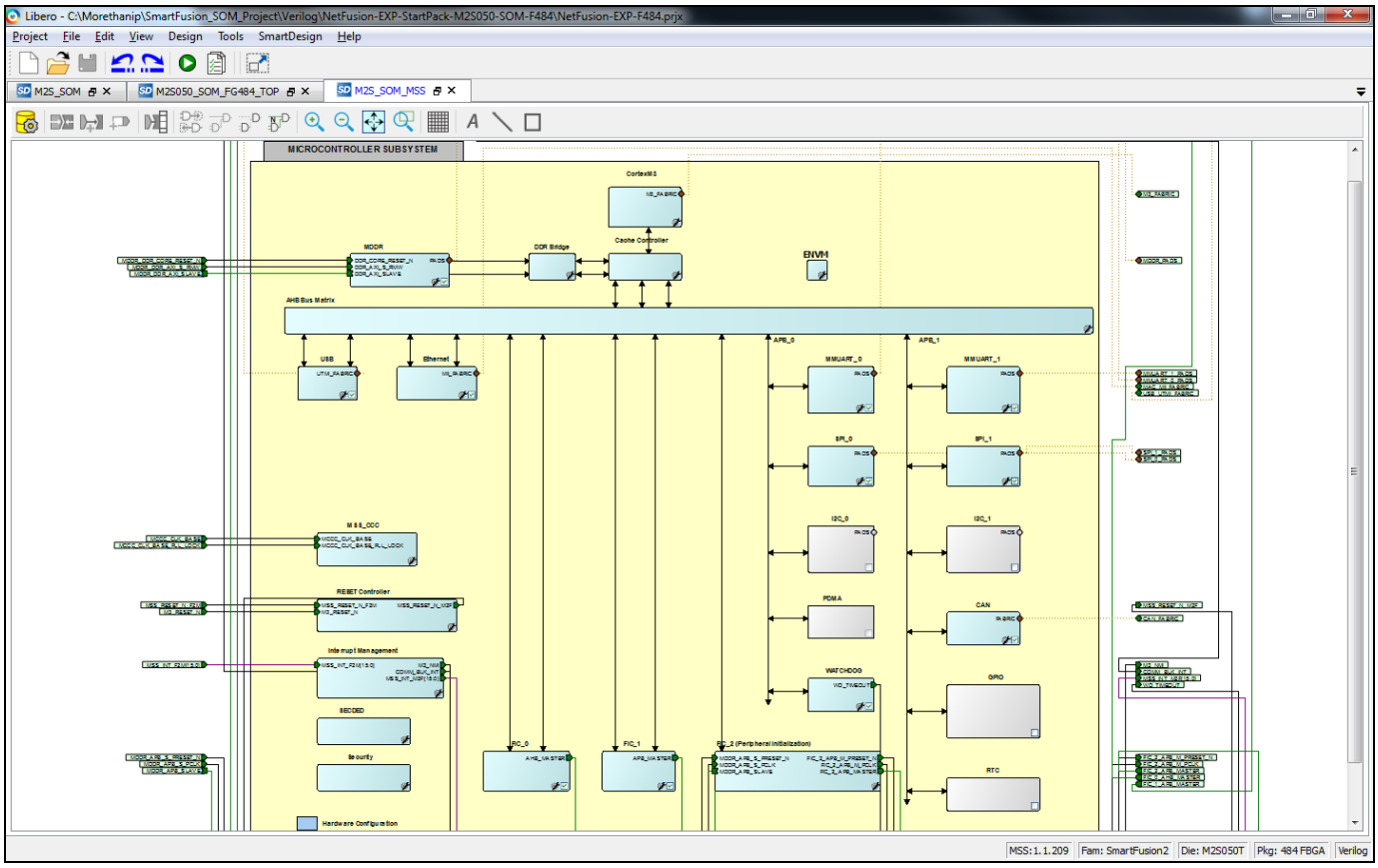


Figure 52 - Exploded View of the Modules in the NetFusion-EXP ARM MSS

The MSS in the heart of the SmartFusion2 NetFusion-EXP design is defaulted and set to the exact current needs of the PCB product. Modules *I2C1, I2C2, GPIO, RTC and PDMA* are disabled currently but there is nothing stopping the user from enabling and wiring out the modules to the SOM sheet.

The blue modules are currently enabled for the NetFusion-EXP design as they have functional requirements and the signals are wired out to the SOM sheet then if necessary up to the I/O assignments then out of the FPGA device itself.

14.1 CAN Controller

The CAN Controller (*shown in the above block*) handles the processing of the data streams in and out of the two transceiver buses. The uCLinux environment is capable of supporting the hardware and is represented as a file descriptor in the typical UNIX way.

14.2 MDDR

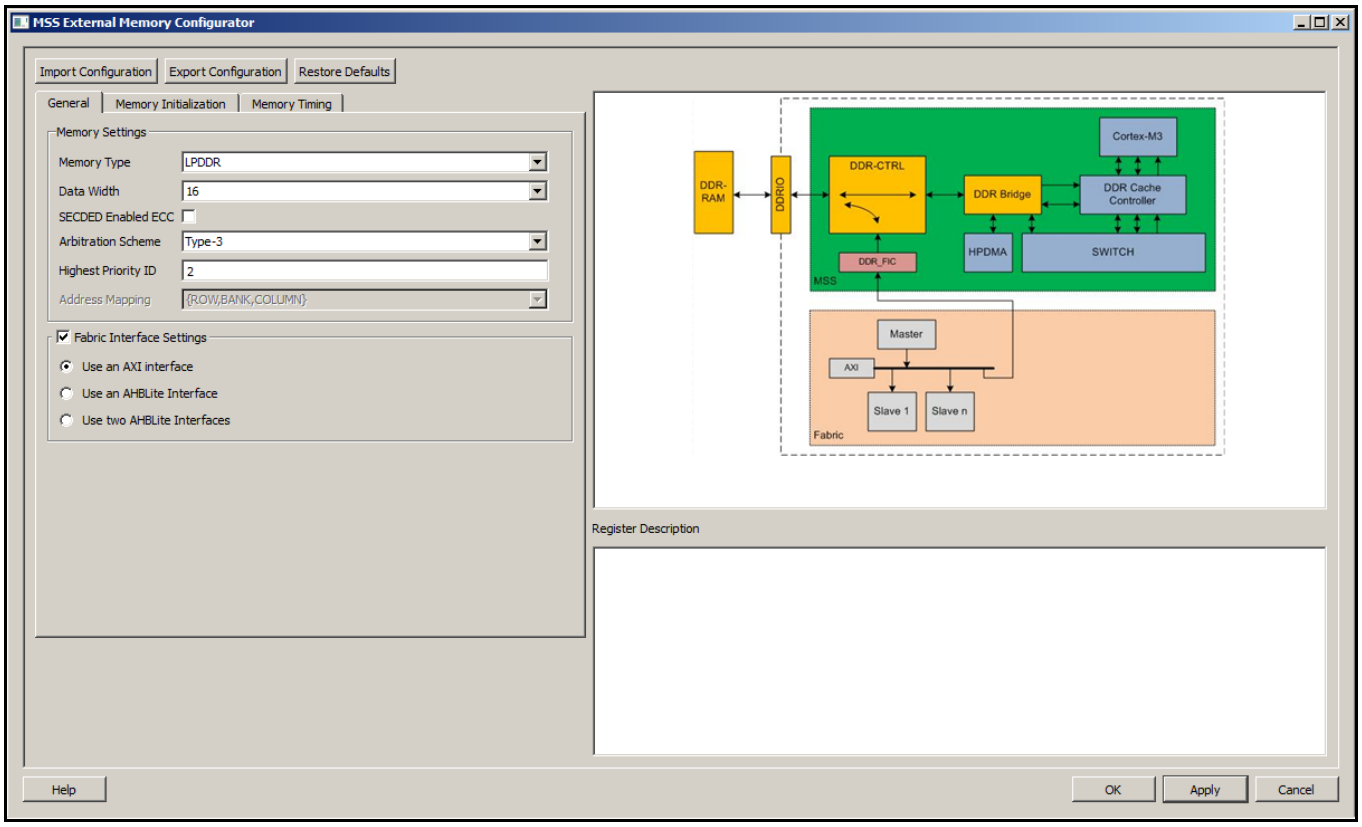


Figure 53 - MDDR MSS Controlling LPDDR Memory with AXI from Fabric

The MDDR has been configured to use a single data rate LPDDR SDRAM device on the M2S-SOM-F484 using 16-bit data width. Priority has been given to the AXI master interface from the fabric where the default connection is to the AMBA DMA Controller for *3rd party MACs and Ethernet SWITCH*.

If you do not wish to keep the DMA Controller or any AXI interface for that matter, then you can de-select the Fabric Interface and the SOM sheet will adjust accordingly. **Important:** *You will have to remove and delete the instantiated DMA Controller however.*

Note: *NetFusion-EXP has to follow the architecture of the SOM-F484 so this is primarily based upon the Emcraft starter project for this block.*

14.3 USB

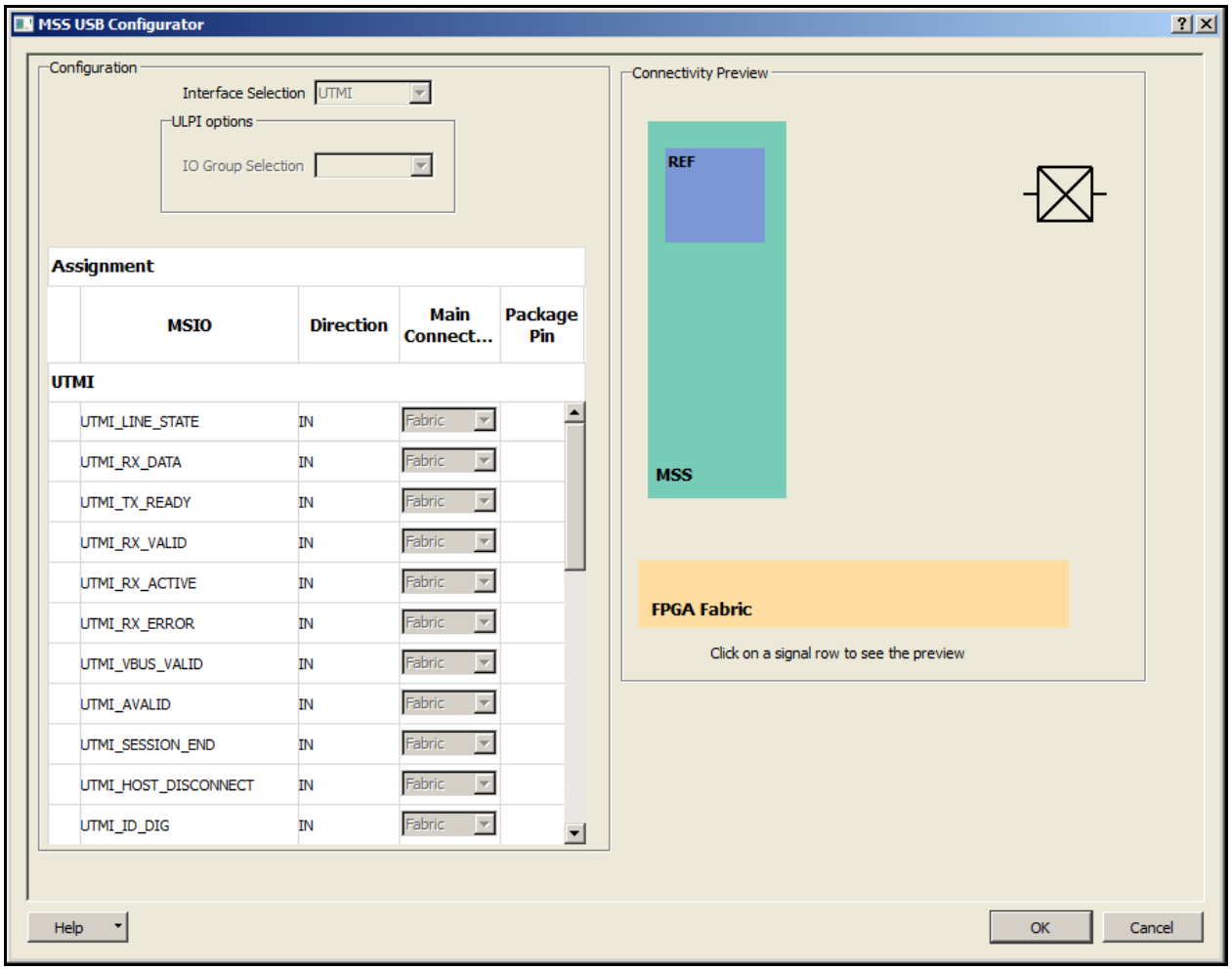


Figure 54 - USB OTG UTMI Host Controller

NetFusion-EXP does utilize on the PCB hardware an *OTG USB* interface. In the fabric, the *ULPI* is converted to UTMI and then connected to this MSS internal block.

The ARM uClinux application code will be able to access this USB block as a block of memory registers and device drivers will be able to control the USB OTG as a USB stack.

Note: *the UTMI signals do not get routed to the I/O pins directly but the fabric for the use of the ULPI/UTMI converter IP core.*

14.4 Ethernet

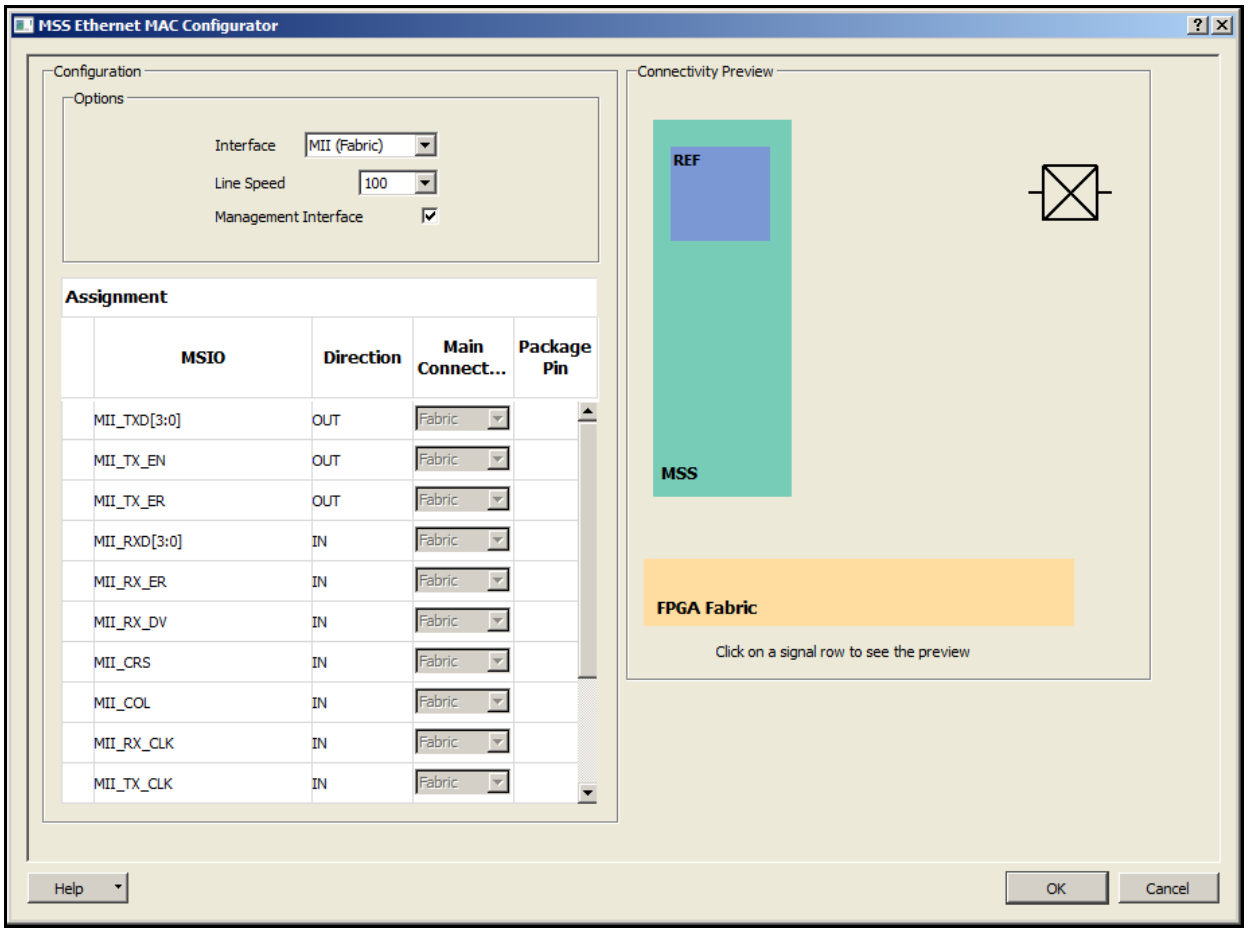


Figure 55 - MSS MAC for 10/100 Ethernet Using SOM PHY

This is a very strategic and important module in the MSS of the SmartFusion2 of NetFusion-EXP. The RJ45 Ethernet connector on the NetFusion-EXP PCB is routed directly to the PHY on the M2S-SOM-F484 which in turn, connects into the SmartFusion2 FPGA and directly into the ARM MSS.

Once in the MSS, the Ethernet MAC receives the connections which then provide a memory interface internally for the ARM uClinux device drivers for the network interfaces.

Note: *The connections route to the fabric before going up through to the I/O assignments.*

It is important to emphasize that with NetFusion-EXP, all of the IO Expander GPIO that routes into the SOM sheet of the starter project and terminates at [*unused_block_0*].

14.5 MSS_CCC

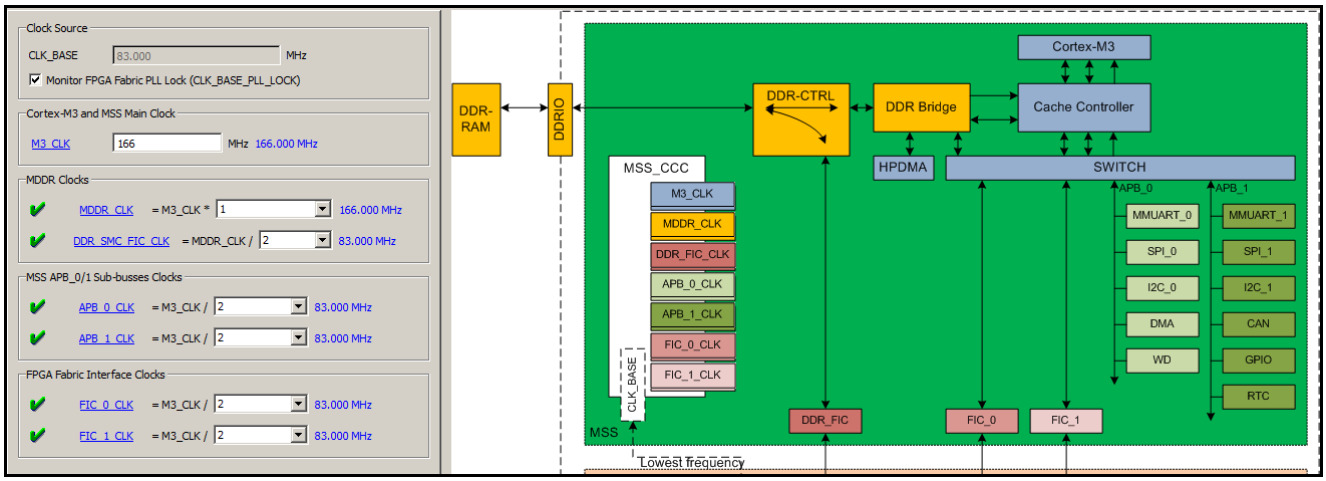


Figure 56 - MSS CCC Divider from the CLK_BASE with-in ARM Sub-System

The main *166MHz* clock into the MSS from the fabric is configured to be split up and if necessary divided down to the different areas of the sub-system. It is derived from the **Div 2** *83MHz* from the PLL in the fabric.

In the case of NetFusion-EXP and the Starter Pack Libero 11.X IDE project, all peripherals, memory controllers and fabric interfaces run at half the base clock frequency at *83MHz*.

14.6 RESET Controller

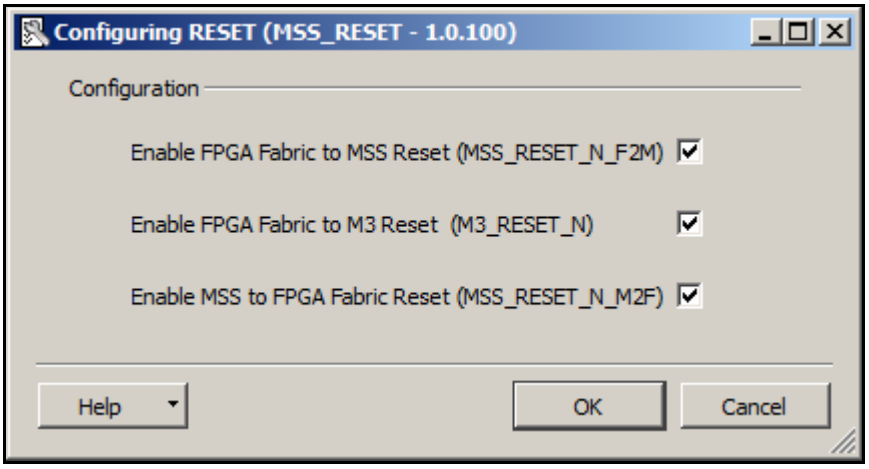


Figure 57 - Reset MSS Module

NetFusion-EXP by default enables the **MSS_RESET_N_F2M**, **M3_RESET_N** and the **MSS_RESET_N_M2F** negative reset signals to come in from the SmartFusion2 FPGA fabric.

This gives more functionality to the user. However, if it is not a required functionality when the design is customized, then they can be de-selected.

14.7 FIC_0

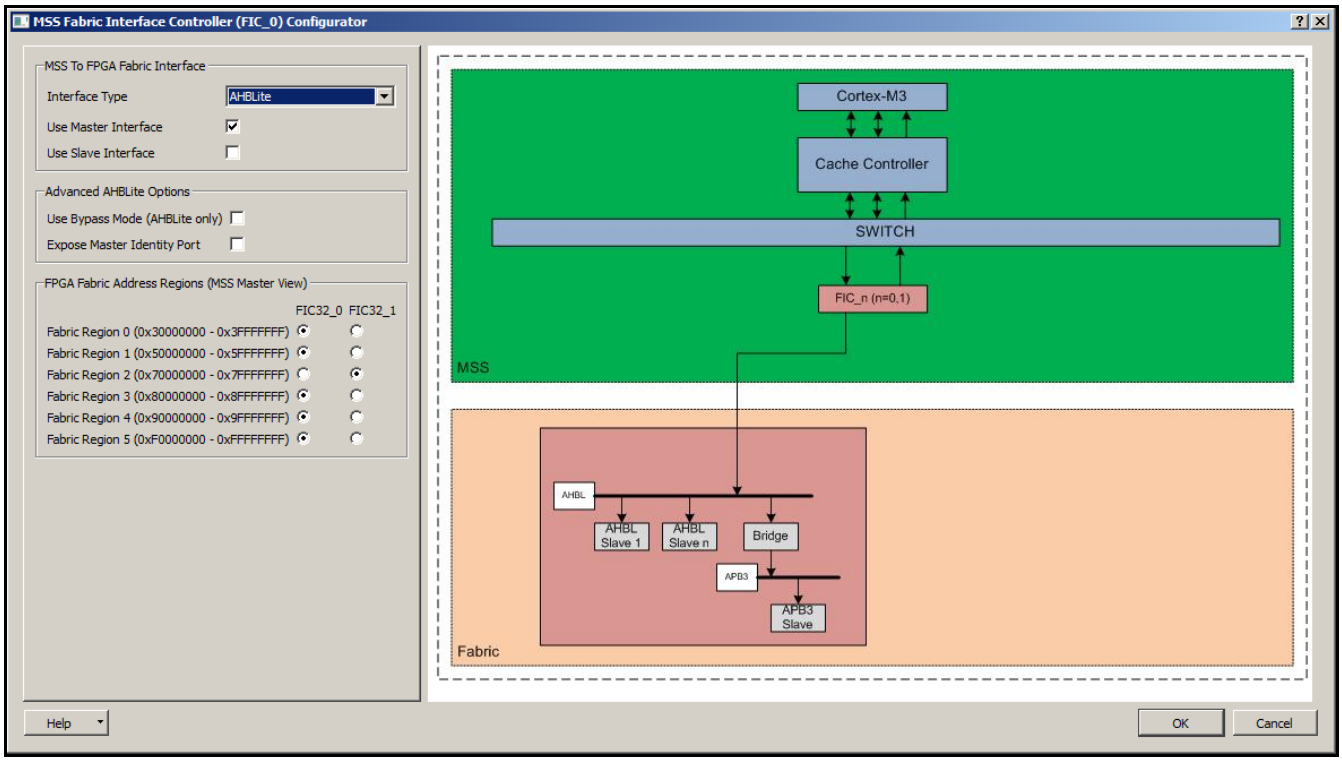


Figure 58 - AHB/APB Fabric Interface [1]

The NetFusion-EXP **FIC_0** has been chosen to be assigned for the AHB-Lite fabric interface. It is a *MASTER* which connects to *the SLAVES* in the fabric as the ARM processor has complete control.

By default, the AHB-Lite only interfaces to the AMBA DMA Controller used for 3rd party Ethernet IP cores. Although the DMA Controller uses *AXI* for the main data throughput, the AHB-Lite is used to access the configuration registers.

Note: *you will observe that Fabric Region 2: 0x7000000 - 0x7FFFFFF has been allocated to the next* ***FIC_1*** *block (next sub-section). The configuration on the left panel (above) selects mapping for both* ***FIC_0*** *and* ***FIC_1****.*

NetFusion-EXP assigns Fabric Region 2 to the APB memory interface so that the *UART, SPI* and *GPIO* can be accessed in the fabric otherwise only this AHB-Lite interface would be mapped which would severely limit the NetFusion-EXP functionality and capability.

14.8 FIC_1

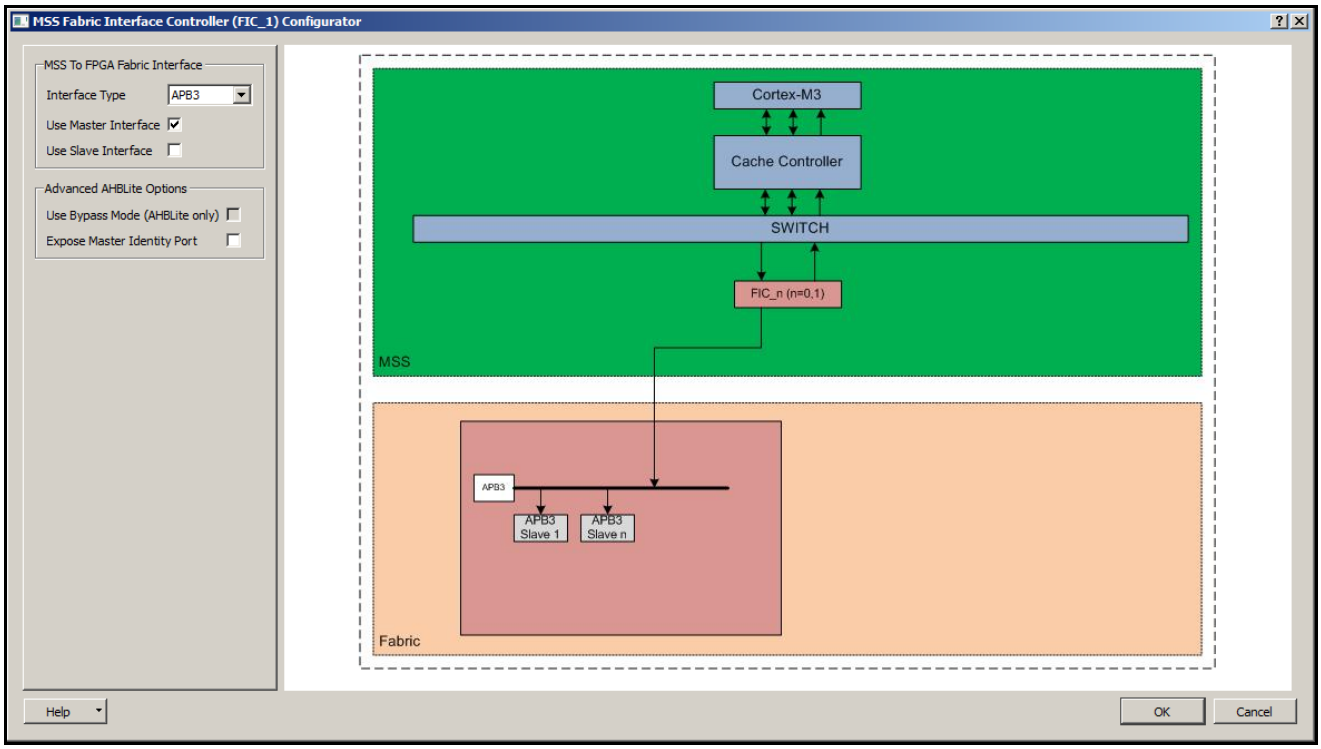


Figure 59 - APB3/AHB-Lite Fabric Interface [2]

Using the configuration from the previous **FIC_0** block *(previous*), this MSS module enables the *APB* interface to access the *GPIO, SPI* and *UART* in the NetFusion-EXP SmartFusion2 fabric.

Note: *this is also an APB Master as the ARM processor has complete control.*

14.9 FIC_2 (Peripheral Initialization)

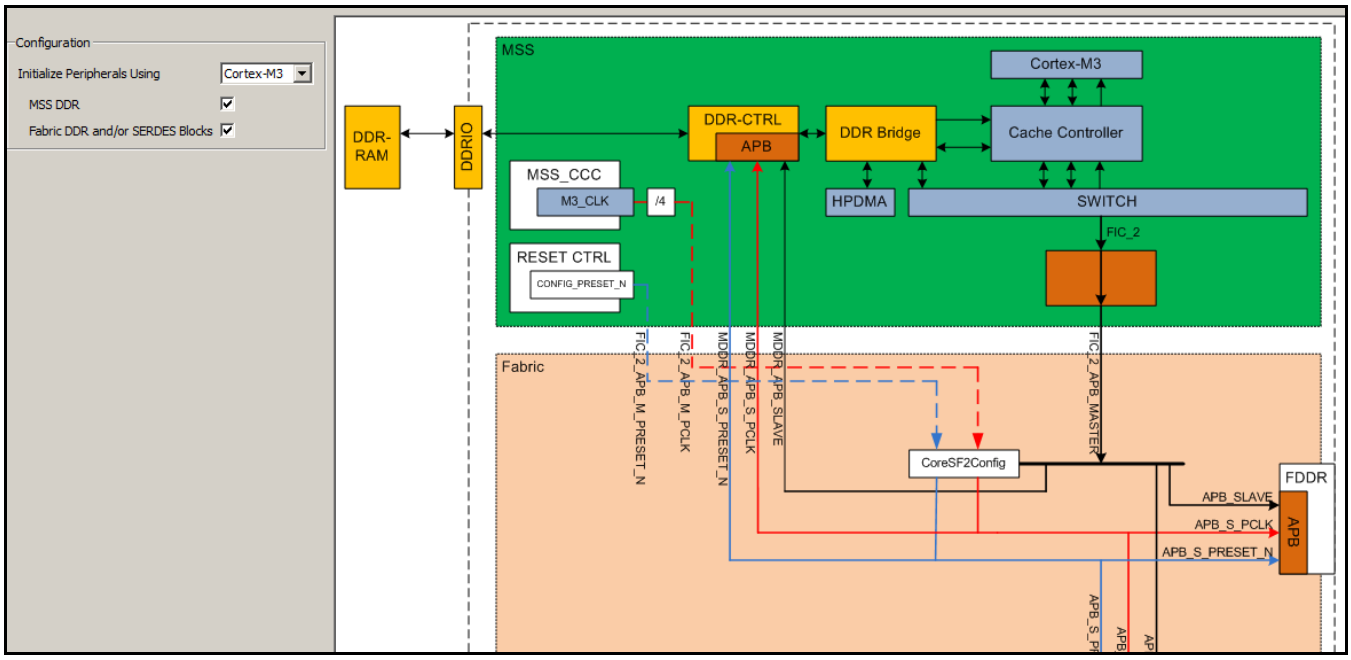


Figure 60 - APB Peripheral Hardware Configuration

As described earlier in this document, the SmartFusion2 FPGA is setup and primarily configured by boot-up software executed by the ARM processor. This can be bare-metal code or early boot code from uClinux called *u-boot*.

This MSS module block enables the signals to connect a feedback *APB* interface from the ARM processor to the other peripherals in the *MSS* itself. It seems overkill but that is the architecture we are ruled by it seems.

In the case of NetFusion-EXP, we configure the **FIC_0** and **FIC_1** fabric interface controllers and also select the AXI Slave interface for the MDDR block.

Important Note: *do not ever remove this feature from the design of NetFusion-EXP. The system will not operate if deleted.*

15 Adding IP Cores from Verilog/VHDL

15.1 Importing Source Files

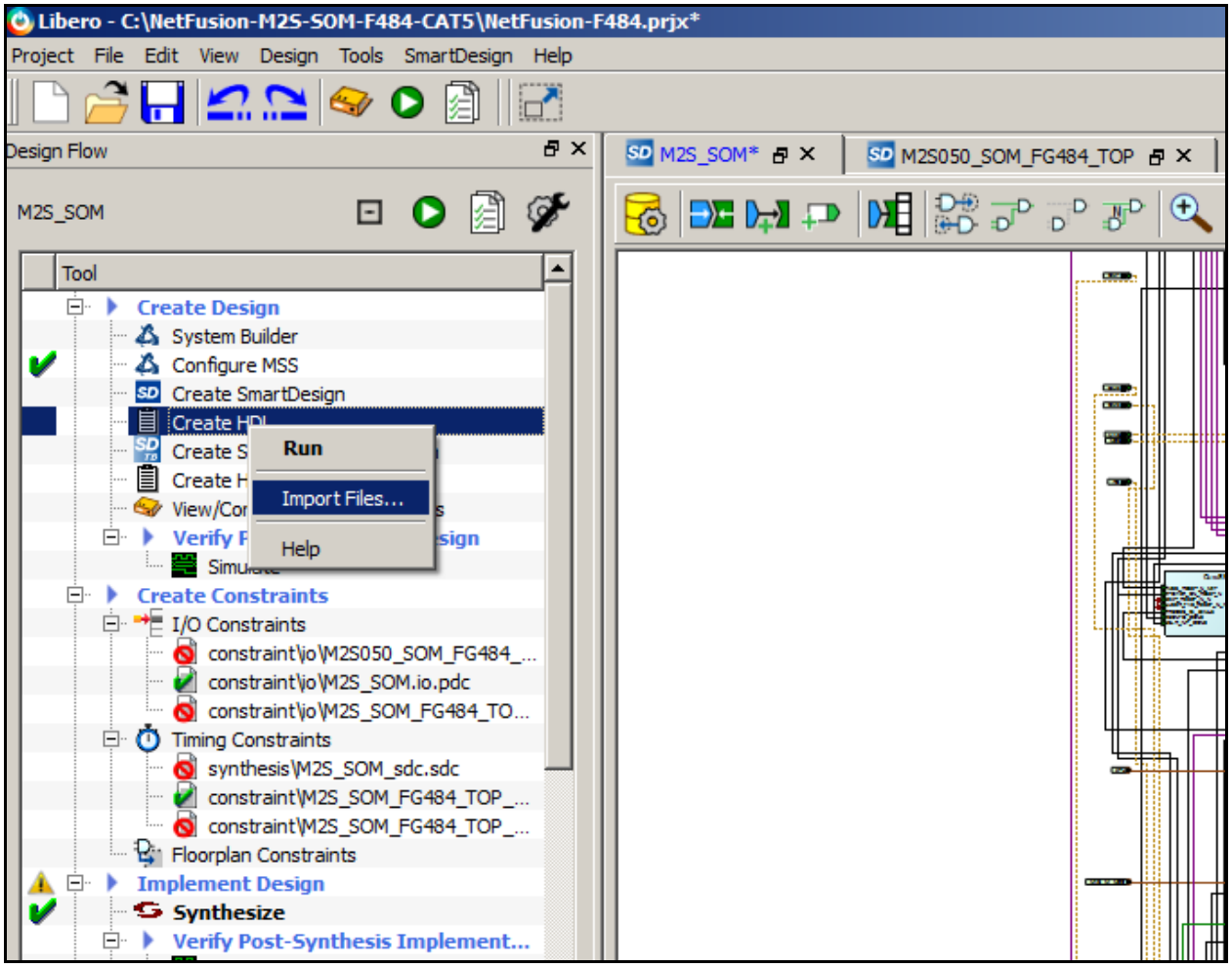


Figure 61 - Importing a single Verilog/VHDL file into the NetFusion-EXP Libero Project

From time to time, you may want to deviate from the default NetFusion-EXP Libero 11.X Starter Project. If you need to add catalogue cores especially for the fabric macros, then download using the vault.

Also, as illustrated *(above*), you also may wish to import 3rd party source IP cores. To do this, on the panel on the left of the Libero IDE, right click on "*Create HDL*" and then select "*Import Files...".* This will bring the Verilog/VHDL into the main system.

You can check if the source code has syntax errors and is valid for the synthesizer (Libero 11.XSynplify Pro) that Libero 11.X uses. This mechanism enables the NetFusion-EXP project to be customized and built on for a user's system requirement.

15.2 Instantiating into the Fabric

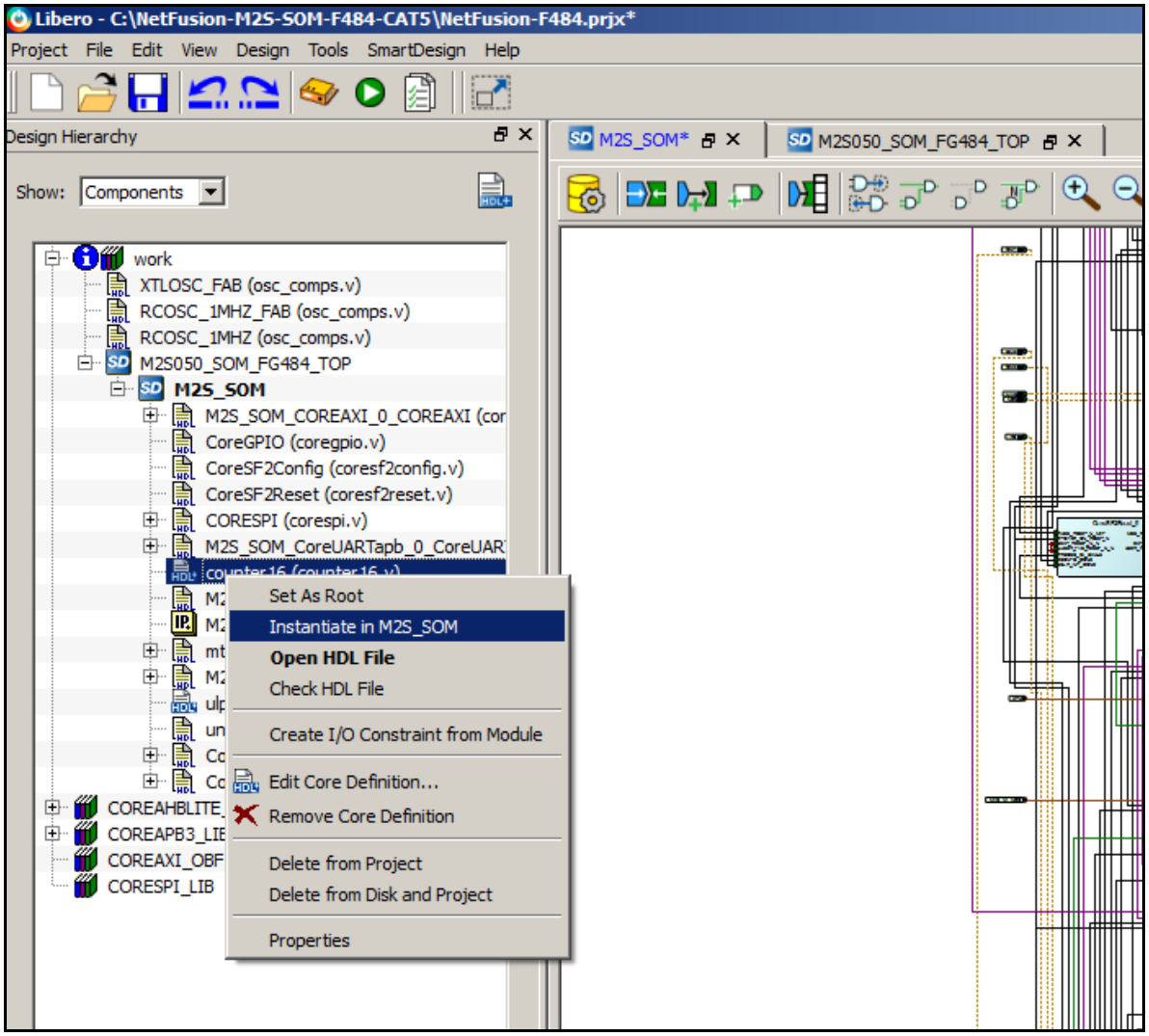


Figure 62 - Instantiating an Imported IP Core in your Fabric Design

Whether you wish to instantiate into the SmartFusion2 fabric an imported 3rd party source code core from VHDL/Verilog or it is from the vault downloaded from ACTEL, the process to get the core into the design sheets is the same and relatively simple. Even if the core is a macro as part of the ASIC area of the FPGA, the process is the same no matter what area of the design it involves. Simply move the left panel to select "*Design Hierarchy*" and then right click on the listed core of your choice. Select *"Instantiate in M2S_SOM"* or whichever sheet is displayed on the right pane of Libero IDE.

The core will appear in the design for you to move and anchor ready for connection routing.

Note: *If the core has errors, the error report page will appear and the instantiation will not occur.*

15.2.1 Building and Synthesizing the NetFusion-EXP Design

- *Open the Top-Level, SOM and the MSS sheets*
- *Start with the MSS and right click "Generate Component"*
- *If errors eliminate then perform previous instruction over again*
- *Once successful, move to the SOM sheet*
- *Right click "Generate Component"*
- *If errors eliminate then perform previous instruction over again*
- *Once successful, move to the Top-Level sheet*
- *Right click "Generate Component"*
- *If errors eliminate then perform previous instruction over again*

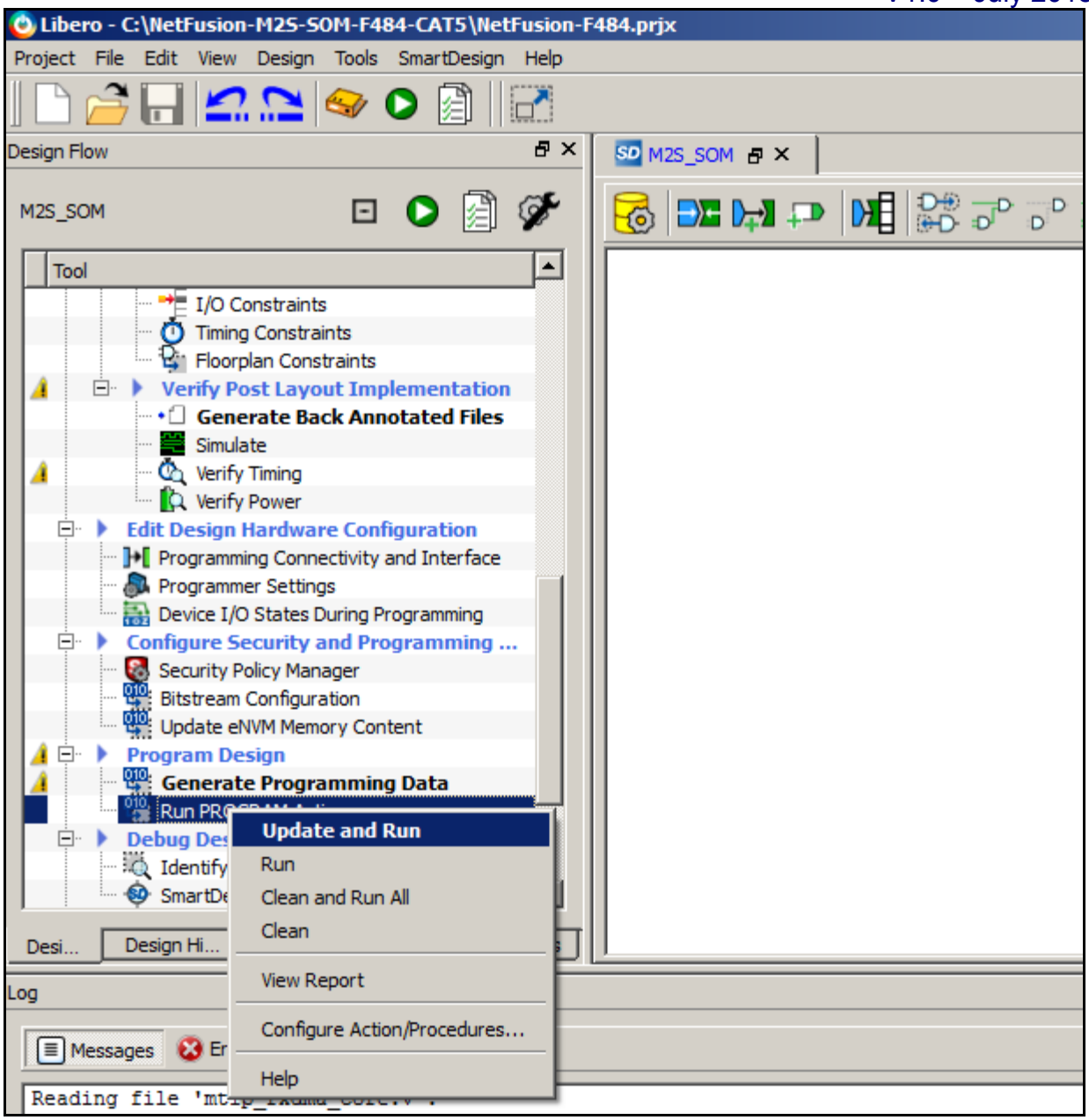


Figure 63 - Selecting Full NetFusion-EXP Synthesis and FPGA Programming

Once you are ready to synthesize and program the SmartFusion2 FPGA on the M2S-SOM-F484 housed on NetFusion-EXP, select the option *(shown above*).

Note: *This should take around 20 minutes on a standard Windows XP/7/8 PC. However, larger customized design will add significant time especially if the design has strict time-constraints.*

16 Using FlashPro4 IDE to Program NetFusion-EXP

Install and make use of *FlashPro IDE* on Windows XP/7/8 OS to allow for the stand-alone programming of the baseboard without the need for launching, installing or using the full **Libero 11.X SoC IDE**.

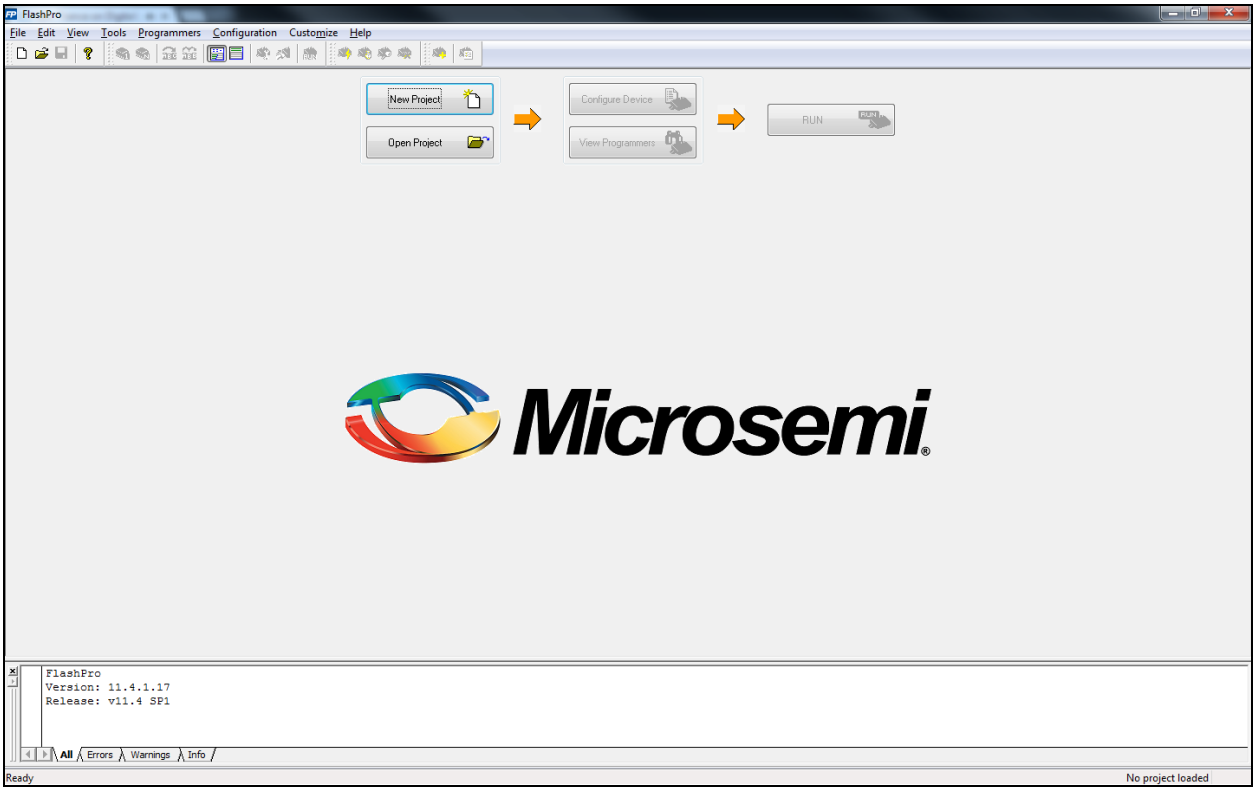


Figure 64 - Opening screen of FlashPro IDE

Launch the IDE so that you see the (*above*) screen. Also, if you have just installed the IDE, ensure that you have also installed the *FlashPro4 USB* device driver in the **Windows Device Manager**.

Note: *the device driver should be automatically installed with the FlashPro installation (which is downloadable from the Microsemi website).*

Ensure that the NetFusion-EXP PCB has the ribbon cable attached from the FlashPro4 USB programmer (*below*)

Figure 65 - Microsemi FlashPro4 ribbon connection

On the FlashPro IDE screen, click on **New Project**.
Enter any project name (*and make sure you have write permissions to the selected directory you choose*).

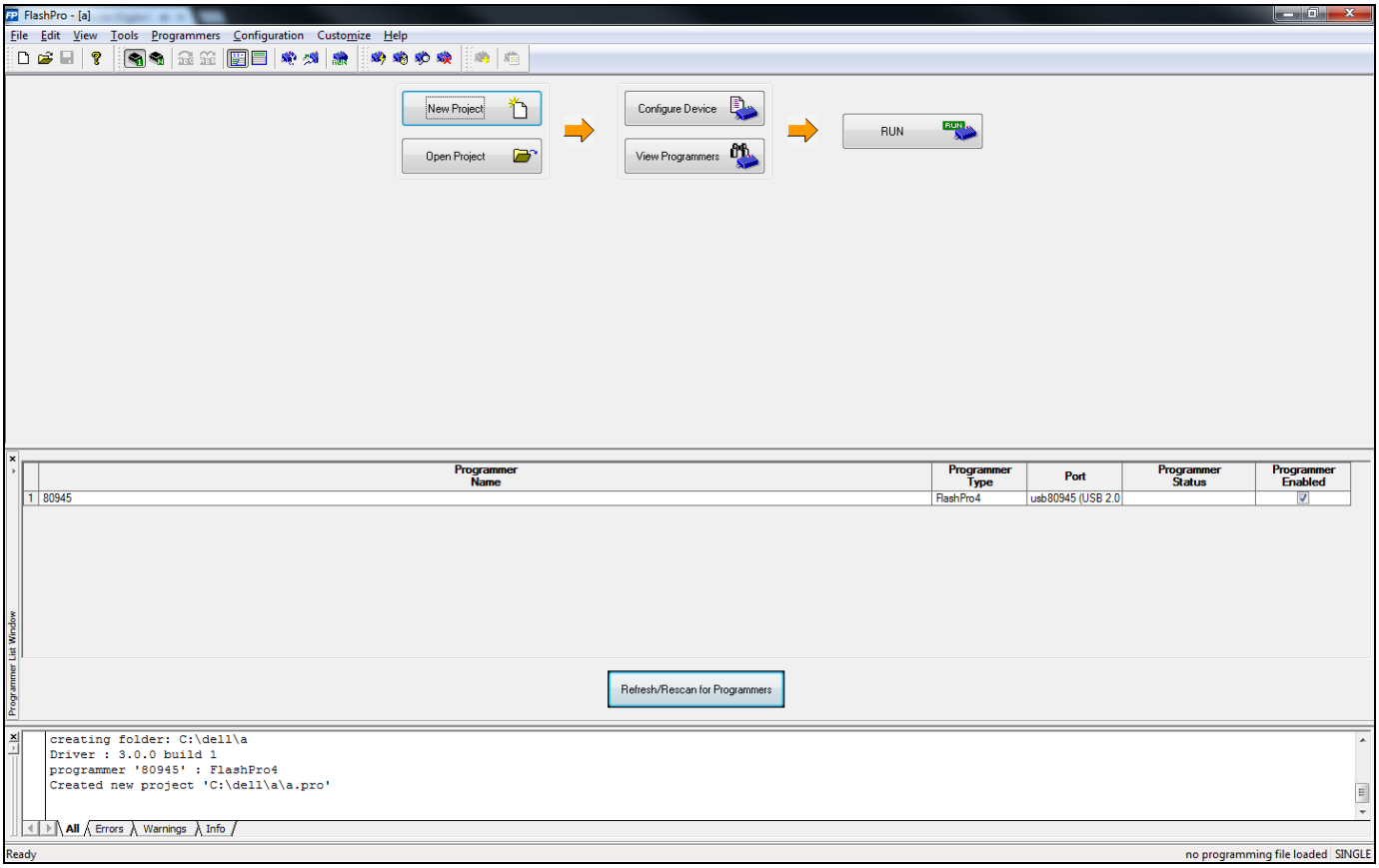


Figure 66 - FlashPro Programming Screen

In the drop-down menus, choose **Configuration**, then select **Load Programming File…**

Chose your **[.STP]** file to program NetFusion-EXP.

Click on the main **PROGRAM** button center right of the (*above*) screen.

Wait for the programming to complete typically after several minutes…

If you cannot select the USB programmer correctly, check it is plugged in and also click on "Refresh/Rescan for Programmers" button towards the bottom of the screen (*above*).

Important Note: *once the operation to program NetFusion-EXP is complete, the SmartFusion2 FPGA will re-power and start automatically.*

17 FAQ

Why even bother with NetFusion-EXP PCB baseboard?

If you want an evaluation PCB, something to develop with before, during or after your own custom PCB development, or because it provides everything you need for your own product idea – NetFusion-EXP is immediately available for use.

All you need to implement is your own software programs, change the uCLinux configuration, write device drivers and maybe if necessary, add in *Libero IDE* more IP cores in the fabric.

Why is the LCD screen not provided as standard?

This can add unwanted cost to the baseboard. As there are so many different manufacturers this is removed from the default build list and the user can choose their own.

Why is the USB OTG connector not available by default out-of-the-box?

As the USB-UART interface allows all of the initial start-up interfacing the user requires, there is no point in providing a second USB. The user can use the OTG interface to suit their own needs once they start to develop their own customization and product development. This is done through the ULPI USB core in the MSS of the SmartFusion2 ARM environment.

Why are the power-supplies over engineered and the interfaces highly clamped and protected?

The voltage regulators and the dc-dc modules are highly designed for noise, transient, surge and ESD suppression. This gives rise to an industrial capability with EM certification which is extremely useful with customer requirements and an added advantage over rival solutions.

Can I use the LCD buttons without the LCD present?

Yes, these are just normal input signals to the SOM GPIO.

What is the battery used for on the PCB?

The baseboard housed Real-Time Clock which can continue maintaining a time for *3 years*. When the power is applied to the NetFusion-EXP PCB, the battery is re-charged slowly up to full charge.

Do we really need such a wide range power voltage input?

This is for the many wide range of power supplies used and already deployed in industry around the world. Obviously, little power is wasted as NetFusion-EXP uses a *dc-dc* module so that a higher voltage uses less current.

What connection topologies does NetFusion support?

As uCLinux is the native processor OS with NetFusion, all *STP, RSTP, SNMP, ICMP* can be used. This will allow for a meshed multi-route system with bridge protocol packets flowing around a NetFusion-EXP connected Ethernet network.

Is a Web server in each NetFusion-EXP the only way to access or change its configuration?

No. A user can access via SSH, USB-UART, serial port, LCD display with buttons, telnet or a range of other proprietary network protocols.

What are the temperature limits for a NetFusion-EXP system?

It confirms to *Industrial Temperature Ranges*. Specifically that is *-40 to +85* degrees Celsius.

What is the maximum serial speed of communication on the serial ports?

230,400 baud

What is the reason for two CAN buses support from the dual transceiver?

Although there is only a single CAN Controller in the SmartFusion2 MSS, using GPIO strobe control, the ARM application uCLinux software applications can select different buses. This is application specific and only a user would know why they would want to use two separate buses. However, it is provided to allow for greater flexibility and support for customers and users to develop with and deploy if needed.

What is the simplest work needed to get a simple working system?

Open the box, attached power supply, power up, attached USB cable to the USB UART, open putty or a PC terminal, plug-in CAT5|6e cable into the RJ45 Ethernet port and link to your PC, press the reset button on NetFusion-EXP, set u-boot IP settings, use u-boot to download new Linux code, reboot, execute!

18 Environmental Specifications

- *Power Supply*: **110/240V** AC @ **50-60Hz** Desk Supply (**31W**) ideally with **18V** DC output connected to **9-36V** wide range Power Receptacle
- *Screw Terminal Alternative Supply Connectors*: **9-36V** DC with maximum current consumption @ **1500mA**
- *Max Power Consumption of Worst Case Operation*: **15W**
- *Operating Temperature Range*: **Industrial**, **-40°C** to **+85°C** degrees
- *EMC/EMI*: **BSEN 61000-6-2:2001 + USA: FCC Part 15 - 2011**
- Non-weatherproof – deployed enclosures must be located inside in an **IP65**-compliant box for industrially mounted situations
- NetFusion-EXP PCB uses Lead-free RoHS and WEEE compliant materials
- *PCB Dimensions*: **170mm** x **170mm** x **1.8mm**
- *Weight (Fully populated with LCD screen & SOM-F484 System-On-Module)* : **1.4kg**

19 Contact

Nine Ways Research & Development Ltd	E-Mail : **pbates@nineways.co.uk** Internet : **www.nineways.co.uk** **UK** Unit A.3, iDCentre, Lathkill House, rtc Business Park London Road, Derby. DE24 8UP United Kingdom Tel : **+44 (0) 1332 258847** FAX : **+44 (0) 1332 258823**

20 Document History

Document Change Notices (DCO)

Version	Description	Created/Changed By	Date
Version 1.0	Initial Release according to Version 1.0	Paul Bates: Nine Ways	**12 July 2015**

Copyright © Nine Ways R&D Ltd 2015. All Rights Reserved.

Table 10: Document History Entry Log